BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

BACHELOR'S THESIS

Brno, 2021 Farhad Abbasi

BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS
ÚSTAV TELEKOMUNIKACÍ

VIRTUALIZATION OF LABORATORY TASKS FOR THE CISCO COURSE
VIRTUALIZACE LABORATORNÍCH ÚLOH PRO KURZ CISCO

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — **Farhad Abbasi**
AUTOR PRÁCE

SUPERVISOR — **Ing. Anna Kubánková, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2021

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Bachelor's Thesis

Bachelor's study program **Telecommunication and Information Systems**

Department of Telecommunications

Student: Farhad Abbasi ***ID:*** 189502

Year of study: 3 ***Academic year:*** 2020/21

TITLE OF THESIS:

Virtualization of laboratory tasks for the CISCO course

INSTRUCTION:

Get acquainted with the materials and laboratory tasks for the new CCNP ENARSI course. Analyze possible simulation environments and then select the most suitable environment in which the tasks will be simulated. Create topologies for each task. For troubleshooting tasks, create .txt configuration files and upload them to the appropriate device in the topology. Design and implement a solution for remote access to prepared tasks.

RECOMMENDED LITERATURE:

[1] Cisco. [online]. [cit. 2019-09-06]. Dostupné z: https://www.cisco.com/

[2] LACOSTE, Raymond, Brad EDGEWORTH. CCNP Enterprise Advanced Routing ENARSI 300-410 Official Cert Guide. Cisco Press, 2020. ISBN-10: 1-58714-525-1. ISBN-13: 978-1-58714-525-4.

Date of project specification: 1.2.2021 ***Deadline for submission:*** 31.5.2021

Supervisor: Ing. Anna Kubánková, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
Chair of study program board

Abstract

This bachelor thesis deals with the laboratory tasks for the new CISCO certification (CCNP ENARSI), get acquainted with the topics, select the most suitable environment for simulating laboratory tasks, then create a topology for each laboratory. In addition to the topology , a configuration file (.txt) was created and uploaded to the appropriate device for troubleshooting labs . Last but not least , a remote access solution was designed and implemented for prepared laboratories.

Keywords

CISCO, Emulation, EVE-NG, GNS3, IOL/U, L2TP/IPsec, Mikrotik, Simulation, VPN, Web-UI

Abstrakt

Tato bakalářská práce se zabývá laboratorními úlohami pro novou certifikaci CISCO (CCNP ENARSI), seznámením se s tématy, výběrem nejvhodnějšího prostředí pro simulaci laboratorních úloh, vytvořením topologie pro každou laboratoř. Kromě topologie byl vytvořen konfigurační soubor (.txt) a nahrán do příslušného zařízení pro úlohy zaměřené na hledání problémů. V neposlední řadě bylo navrženo a implementováno řešení vzdáleného přístupu pro připravené laboratoře.

Klíčová slova

CISCO, Emulace, EVE-NG, GNS3, IOL/U, L2TP/IPsec, Mikrotik, Simulace, VPN, Web-UI

Rozšířený abstrakt

Společnost Cisco spustila nové certifikace 24. února 2020. Hlavní struktura zůstává (Entry, Associate, Professional, Expert), ale certifikační cesty a obsah každého certifikátu byly změněny a upraveny. Nové certifikace se skládají ze sedmi nových směru: Enterprise, Collaboration, Data Center, Security, Service Provider, CyberOps a DevNet. Na úrovni associate má společnost Cisco pouze tři směry zakončené certifikační zkouškou a to Enterprise, CyberOps a DevNet. Na profesionální úrovni se každá cesta skládá z jedné základní zkoušky (Core exam) a alespoň dvě volitelné zkoušky. Chcete-li však získat certifikaci Cisco Certified Network Professional (CCNP), musíte složit základní zkoušku s jednou volitelnou zkouškou. Za zmínku stojí, že základní zkouška je také předpokladem pro odbornou úroveň (CCIE). Enterprise pro CCNP se skládá z jedné zkoušky Core (350-401 ENCOR) a šesti volitelných zkoušek, z nichž jednou je Cisco Enterprise Advanced Routing and Services Implementation (300-410 ENARSI).

Tato bakalářská práce se zabývá laboratorními úlohami pro novou certifikaci CISCO (CCNP ENARSI), seznámením se s tématy, výběrem nejvhodnějšího prostředí pro simulaci laboratorních úloh a vytvořením topologie pro každou laboratoř. Kromě topologie byl vytvořen konfigurační soubor (.txt) a nahrán do příslušného zařízení pro úlohy zaměřené na hledání problémů. V neposlední řadě bylo navrženo a implementováno řešení vzdáleného přístupu pro připravené laboratoře.

Na začátku je technicky rozlišen význam dvou použitých slov vzájemně zaměnitelných: „Simulace "a „Emulace ". Účelem této práce je emulace laboratorních úloh. Emulátory by měly běžet v nejvhodnějším prostředí. Dvěma nejběžnějšími prostředími, která emulátory převážně podporují, jsou Graphical Network Simulator-3 (GNS-3) a Emulated Virtual Environment - Next Generation (EVE-NG). GNS-3 je open-source software a EVE-NG má také jeho bezplatnou verzi (Community). Jako nejvhodnější prostředí bylo vybráno EVE-NG, protože GNS 3 obsahuje chyby (bug) a méně uživatelsky přívětivé webové uživatelské rozhraní a má nestabilní spojení s instalací GNS3 na vzdáleném serveru, který používá software OpenVPN.

Byly testovány různé emulátory, například Dynamips (Cisco IOS 3725 a Cisco IOS 7206VXR), Qemu (CISCO CSR 1000v (XE 3.x)), ale nepodporovaly všechny příkazy pro laboratorní úlohy. Z důvodu nedostatečné podpory bylo rozhodnuto použít IOS On Linux/Unix (IOL/U) pro zařízení L2/L3 a Qemu pro Windows hostitele. IOL/U je produkt pouze pro CISCO, je určen pouze pro zaměstnance CISCO a k jeho používání je potřeba licenci. Podle doporučení EVE-NG byly použity tři různé verze obrazu IOL a pro všechny obrazy byla vytvořena jedna licence.

Laboratorní úlohy se skládají ze dvou typů: Implementace a Troubleshooting, což je celkem 40 úloh. Implementace zahrnují topologii sítě, počáteční konfiguraci pro každé zařízení a různé související konfigurační úlohy. Troubleshooting zahrnují jeden nebo více ticketů, topologii sítě a konfigurace zařízení pro každý tiket. V rámci EVE-NG byla nakreslena topologie, konfigurace byly upraveny (pouze typ a počet rozhraní) a nahrány do každého příslušného zařízení, poté byl vytvořen konfigurační soubor (.txt) a uložen na základě startup-config každého zařízení.

Nejzajímavější částí této práce je návrh a implementace řešení vzdáleného přístupu pro laboratorní úlohy připravené na EVE-NG. Tento problém byl vyřešen implementací řešení L2TP/IPsec VPN. Řešení je založeno na modelu klient-server, na straně serveru se skládá z EVE-NG nainstalovaného na pracovní stanici Dell a routeru Mikrotik jako server L2TP/IPsec, na straně klienta stačí pouze jakýkoli podporovaný operační systém L2TP/IPsec pro přístup k prostředkům serveru.

ABBASI, Farhad. Virtualizace laboratorních úloh pro kurz CISCO. Brno, 2021, 124s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Anna Kubánková, Ph.D.

Declaration

I declare that I have written this Bachelor's Thesis on the theme of "Virtualization of laboratory tasks for the CISCO course" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis.

As the author I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation S 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

In Brno: …… Author's signature: ……

Acknowledgment

In loving memory of my mother, she will always hold a special place in my heart. I would like to express my gratitude towards my family, and especially my brother Jawad Abbasi for his ongoing support and encouragement. I would like to express my special gratitude and thanks to my supervisor Ing. Anna Kubánková, Ph.D. for giving me such attention and time, she has guided me and supervised me to reach this level of capacity. I feel proud of my achievements, it is all thanks to everyone who helped me on my journey.

In Brno: …… Author's signature: ----------

TABLE OF CONTENTS

FIGURES

TABLES

INTRODUCTION

The following bachelor thesis deals with materials and laboratory tasks for the new CISCO certificate, CCNP Enterprise Advanced Routing and Services (ENARSI). The main idea of this thesis is to familiarize with the topics, analyze possible simulation environments and then select the most suitable environment in which the laboratory tasks will be simulated and also create topologies for each task and configuration files (.txt) for troubleshooting tasks, then upload them to the appropriate device in the topology. The last task of this thesis is to design and implement a solution for remote access to prepared tasks.

The first chapter starts with differentiating two words "Simulation" and "Emulation" in technical terms. Then various types of emulators are described and the use of GNS3 and EVE-NG environments for emulation of laboratory tasks are analyzed. The EVE-NG environment has been selected as the most suitable environment, as GNS3 Web-UI (BETA version) is buggy and less user-friendly and has an unstable connection with the GNS3 installation on the remote server that uses the OpenVPN service.

The second chapter deals with preparing the EVE-NG environment for emulating laboratory tasks, instructions about how to upload the emulators to the EVE-NG server, and setting them up.

The third chapter explains the remote access solution to the prepared laboratory tasks. This problem will be solved by implementing the L2TP/IPsec VPN, this is a VPN service between the L2TP-supported Operating System (OS) as an L2TP Client and Mikrotik router as an L2TP Server. This solution is based on the Client-Server model, that's why this chapter is divided into two parts to describe configurations of each part of the Client and Server in detail.

Each chapter from 4 to 15 is related to one of the topics of the ENARSI certification exam, some configuration commands are explained with the help of the emulated "Implementation Labs", and seven trouble tickets have been worked out and explained for different topics.

In conclusion, a summary of the work that has been done in this thesis is given.

1. NETWORK SIMULATION

The two words “Simulation” and “Emulation” of a system are used interchangeably, but in technical terms, they have different meanings: When something is simulated, it means that one system works similarly to another, but its implementation is completely different. In a simulation, we repeat the basic function of the system, but we do not necessarily follow all the rules of what we simulate. Using the simulator, we can get a good idea of how a system works [4]. The most useful network simulators are Network Simulator (NS), OPNET, and Cisco Packet Tracer. On the other hand, emulating a system is trying to work like that system in all aspects. An emulator replicates every single function of the system as much as possible until it gets an exact copy of that system [4]. Graphical network simulator-3 (GNS-3) and Emulated Virtual Environment – Next Generation (EVE-NG) are the two most common environments that mainly support the emulators. It’s worth mentioning that GNS-3 supports both network simulators and emulators [41]. In this thesis, we are looking for emulation of the laboratory tasks, not the simulation. This is because, network simulator, unlike an emulator, can't provide the same experience as working with a real device, as an example, doesn't support some command lines. In this chapter, various types of emulators will be described and the use of GNS3 and EVE-NG environments for the emulation of laboratory tasks will be analyzed.

1.1 Emulators

An emulator is a software or hardware that allows a computer system (host) to use software, peripherals, and tools of another computer system (guest). The emulator enables the host to behave like a guest [42]. The following emulators are the common types of emulators that are supported by GNS3 and EVE-NG:

1.1.1 Dynamips

Dynamips is a Cisco router emulator that supports Cisco platforms such as 1700, 2600, 2691, 3600, 3725, 3745, and 7200. Dynamips emulates by directly booting a real Cisco IOS image and has been supported by Linux, Mac OS X, or Windows operating systems [43].

1.1.2 IOL/IOU

IOS on Linux (IOL) or IOS on Unix (IOU) is a Cisco internal emulator that Linux version is compiled for i386 architecture and the Unix version is compiled for Sparc architecture. This emulator is used by Cisco staff or authorized customers, which is required an iourc license to be run (officially released by Cisco or it can be found on the internet) [44]. IOL/IOU images on the internet are not officially released by Cisco and that is why has not been recommended because it can be found buggy. It supports both L2 and L3 Cisco images and not CPU and memory resource-intensive, which makes it a good choice [41].

1.1.3 QEMU

Qemu is an open-source emulator which can be found in two operation modes of machine emulator and virtualized [45]. Most of the modern appliances such as Cisco's Virtual Internet Routing Lab (VIRL) images (ASAv, NX-OSv, XRv, IOSv, IOSvL2) run on Qemu. Qemu and IOL/IOU both need Nested Virtualization which is not supported by Windows and must be run on Linux.

1.1.4 VPCS

It is a light personal computer emulator. The PC doesn't support GUI but only some simple commands such as ping, traceroute, IP config, and enables to assign an IP address via Dynamic Host Configuration Protocol (DHCP) [41].

1.2 Graphical network simulator-3 (GNS3)

GNS3 is free and open-source software that has been developed by Jeremy Grossman. GNS3 has been created originally for emulating Cisco equipment with Dynamips emulator, but nowadays it supports different emulators for different vendors such as Juniper, Mikrotik, Fortinet, Brocade vRouters, Docker instances, HPE VSRs, Cisco ASAs, and Cisco virtual switches. GNS3 is installed on a local PC (Windows, Linux, and Mac) using the GNS3-all-in-one software (GUI). It can run appliances on a local server (on the same host), in a virtual machine (by downloading and installing the GNS3 VM which is the more recommended option), or run appliances on a remote server (by the installation of the OpenVPN software on both server and client-side). The GNS3 VM is often recommended because it is based on Linux, supports nested virtualization which makes an ideal environment to emulated big-size and advanced topologies equipped with Cisco images (IOSvL2/L3, ASAv) or IOL/IOU emulators [41].

1.2.1 GNS3 graphical environment

The following Figure 1. 1 shows the graphical environment for GNS3 which consists of six main parts [41]:

Workspace: The “dragged and dropped” place for nodes.
Toolbar: The Toolbar is located at the top of the GNS3 GUI which includes icons that perform the tasks such as creating/opening a project, writing notes, drawing custom shapes, Start all nodes, etc.
Device toolbar: In the Device toolbar, the devices are categorized into different types of Routers, Switches, End devices, Security devices, and the Link icon is also located there to interconnect these devices.
Server summary: Server Summary shows the currently used servers along with their resource utilization.
Topology summary: The Topology server provides a list of nodes that have been added to the Workspace.
Console Pane: The pane at the bottom of the page displays any errors/issues messages.

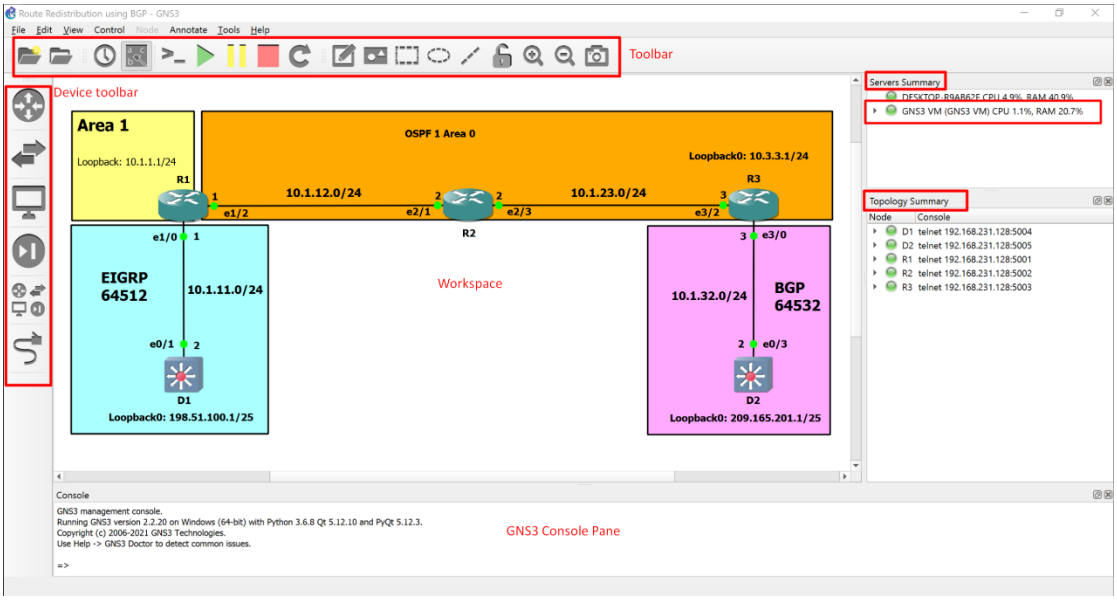


Figure 1. 1: GNS3 Graphical User Interface (GUI) for PC clients

Note = On the 7th of July 2020, the GNS3 team announced that a beta version of the GNS3 Web-UI has been released. The GNS3 Web-UI requires a Web client Pack to be installed on the client machine. Additionally, they have plans for HTML5 console support. To run GNS3 Web-UI, we need to download the GNS3 VM (ova file) and import it to the hypervisors such as VMware Workstation, Player, etc, that is it! We can get access the GNS3 Web-UI via web browsing with the provided management IP address or through the GNS3 GUI installed on the local PC [41], [46]. The following Figure 1. 2 shows the GNS3 Web-UI with the same topology designed in the GNS3 desktop client:

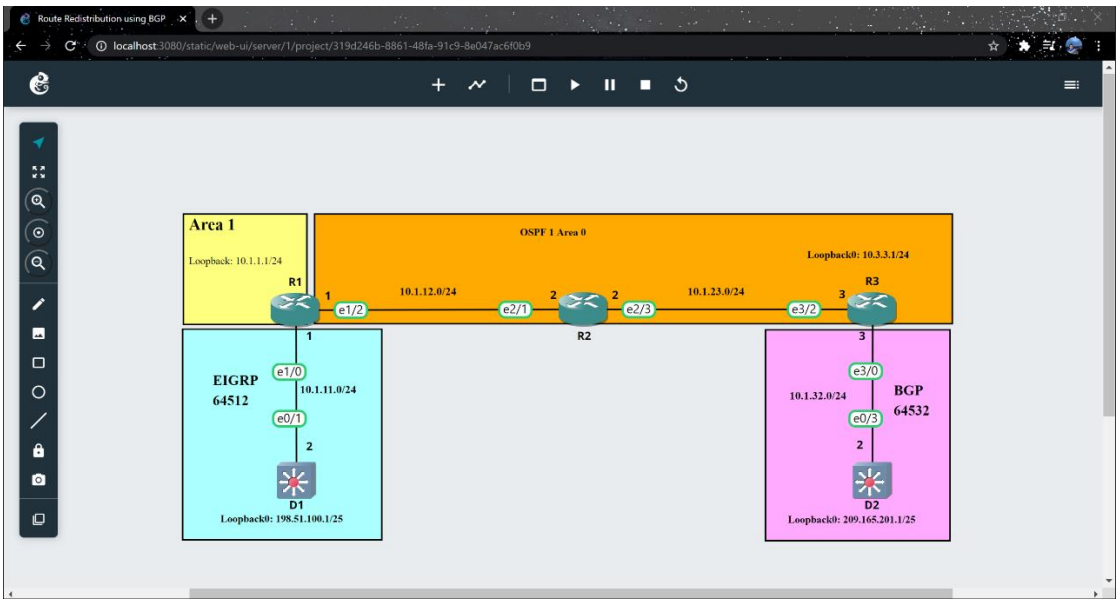


Figure 1. 2: GNS3 graphical web environment

1.2.2 Uploading the emulator images

There are two ways to upload the emulator images inside the GNS3 [41]: **1)** In the Menu bar Edit > Preferences, it only needs to select the type of emulator on the newly loaded window, then the emulator image will be added by clicking the “New button”, see Figure 1. 3, **2)** download the GN3 appliance (prepared template for importing the images), then import the appliance via File > Import appliance in the Menu bar. On the newly loaded window, the user will be asked to select the version of the image for that appliance. If the image has been located inside the local PC, then GNS3 automatically finds the image in the directory otherwise it is needed to download it separately.

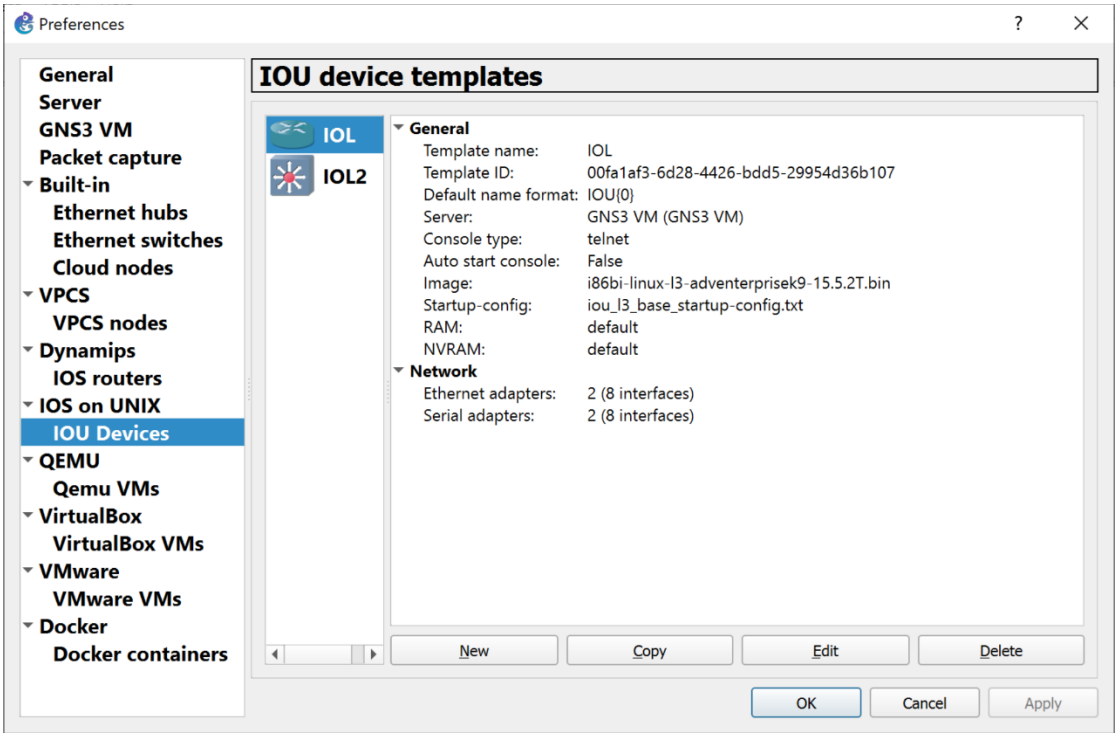


Figure 1. 3: Two IOL images have been uploaded in GNS3 via Edit > Preferences in the Menu bar

1.3 Emulated Virtual Environment – Next Generation (EVE-NG)

EVE-NG is the first clientless network emulation software. The clientless feature allows the EVE-NG to be run in a fully isolated environment [48].

EVE-NG is found in three versions of Community, Professional, and Learning Center, the two last versions of Professional and Learning Center require buying a license [47]. All the versions support the mentioned emulators in section 1.1:

The EVE-NG Community version is free, this version provides an admin role for the user and supports local Wireshark capture, local Telnet, and VNC client. The number of nodes is limited to 63 per lab. The community version is more time-consuming and less comfortable but it is sufficient for the requirements of this thesis [47].

The EVE-NG Professional version provides an admin role for users like the Community version. It supports Export/Import configs to local PC, local Telnet and VNC client, multi Radius servers for user authentication, Docker container, Docker integrated Wireshark, NAT Cloud, and advanced Lab design objects. The number of nodes is limited to 1024 per lab [47].

The EVE-NG Learning Center version has been designed mainly for study purposes which provide an independent role of admin, user, and editor for users. For example, students can join the prepared labs with an independent role under the control of the teacher (Admin). The Learning Center version of EVE-NG supports all the features in the Professional version but the only big difference is the limitation of CPU/RAM per user.

1.3.1 EVE-NG installation

The EVE-NG is a virtual machine that has been released as an OVF file and ISO. It can be installed on top of a hypervisor, physical hardware, and Google cloud (Official supported by EVE-NG) [44],[48]:

Hypervisor Install: The EVE-NG can be installed on top of a virtual machine platform by importing its OVF file (Pre-built virtual machine) either ISO file (Custom installation of a virtual machine). EVE-NG officially supports VMware products such as ESXi (version 6.0 or later), Workstation (14.0 or later), Fusion (8 or later), and Player (14.0 or later).

Bare Install: Because the EVE-NG runs multiple hypervisors and the nested virtualization makes down the performance of the lab, it is mostly recommended a bare installation of EVE-NG on a dedicated physical server (requirement: Support of Intel VT-x, e.g., Intel Xeon CPU) by using the ISO file.

Google Cloud Install: There is also another way of running EVE-NG which is the deployment of the EVE-NG Pro version on a cloud platform such as Google Cloud Platform (officially supported).

1.3.2 EVE-NG graphical environment

The EVE-NG Graphical User Interface (GUI) is available only via web browsing with the provided management IP address (via DHCP or static during the installation of the EVE-NG), see Figure 1. 4 The EVE-NG supports two types of management console: A) Native console, B) Html5 console. The Native console allows to use the native applications such as the Putty and UltraVNC to access the console of the lab nodes. It is recommended to download and install the EVE-NG software pack on the client-side. On the other hand, the Html5 console provides a clientless solution that is not restricted to any terminal such as Telnet or SecureCRT. The nodes can be managed directly through the browser by opening a new browser window [48].

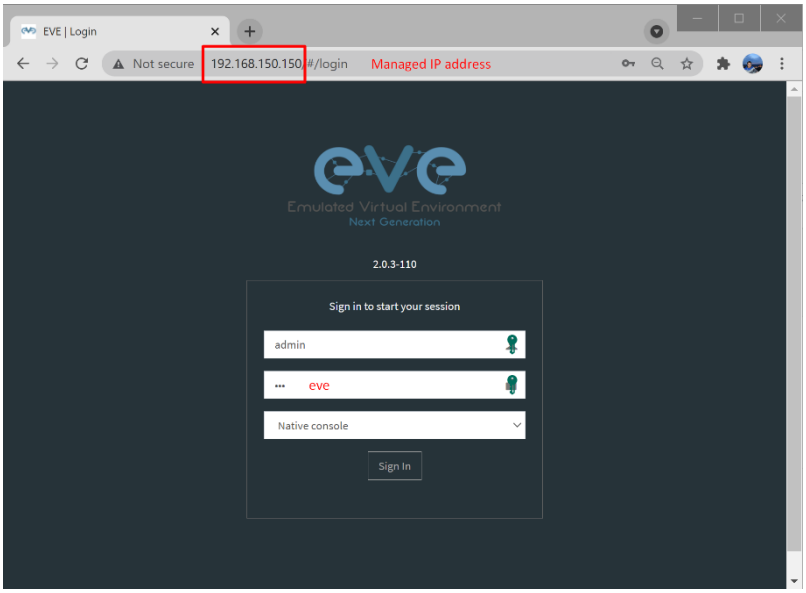


Figure 1. 4: EVE-NG graphical web environment: The EVE-NG login page

The following Figure 1. 5 shows the main EVE-NG management window after it gets signed into EVE:

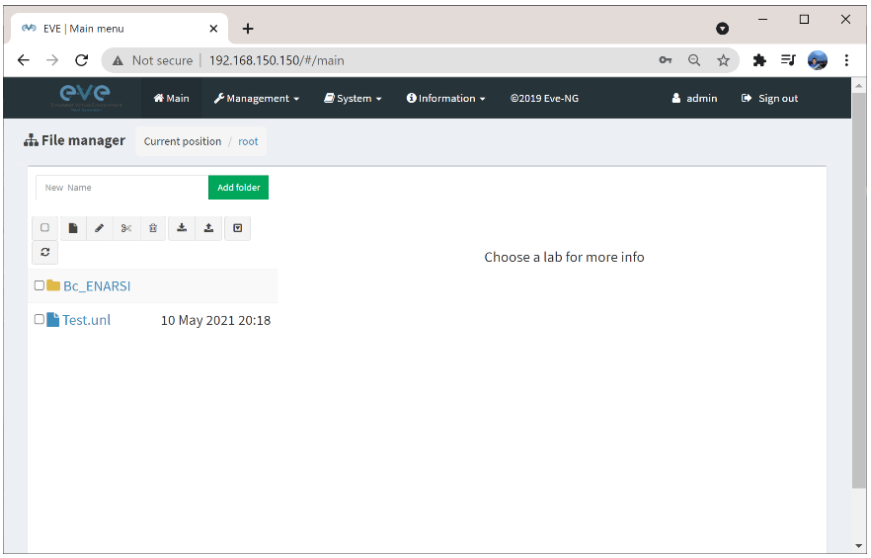


Figure 1. 5: EVE-NG graphical web environment: The Main EVE management window

The "Management buttons" is located at the top left side of this Window where labs can be added, deleted, imported, and exported, see Figure 1. 6:

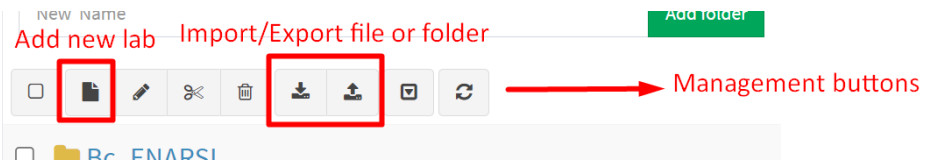


Figure 1. 6: EVE management buttons

The following Figure 1. 7 shows the Topology page when a lab gets open:

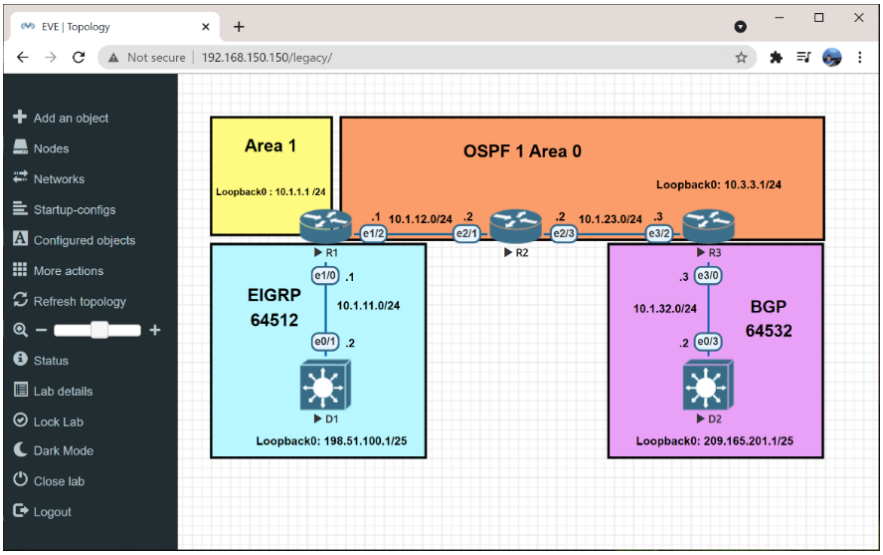


Figure 1. 7: EVE-NG graphical web environment: The Topology page for the opened lab

A brief explanation of functions for the sidebar menu on the Topology page [48]:

Add an object: This icon provides us to add a node, network (bridge, cloud, and NAT in Pro version), custom shape, picture, and text.
Nodes: The Nodes object provides an overview of configured nodes and the possibility of editing them.
Networks: The Networks object provides an overview of configured networks (bridge, cloud, and Nat in pro version) and the possibility of editing or deleting them.
Startup-configs: The Startup-config object provides an overview of the startup configuration of nodes and the possibility of editing them.

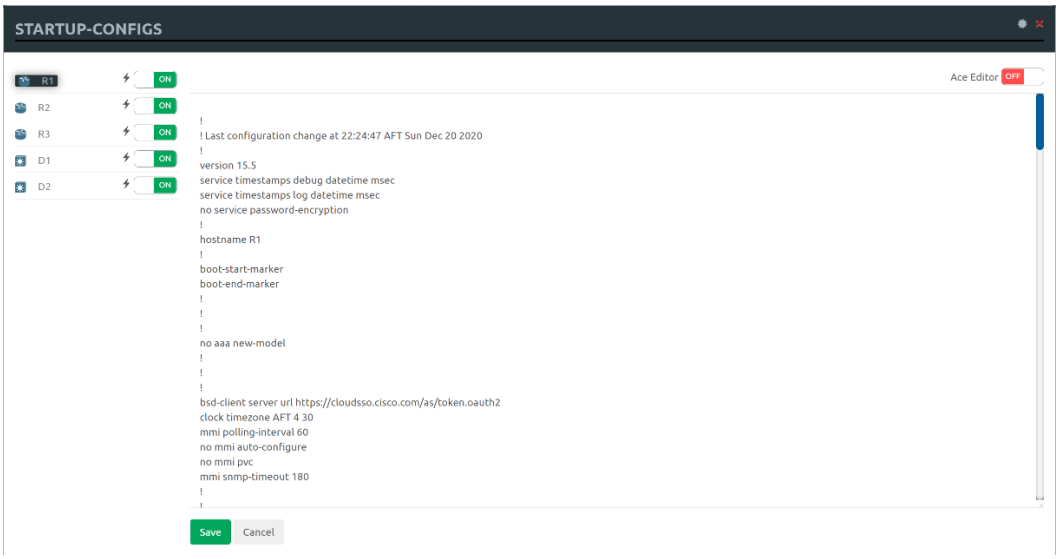


Figure 1. 8: An overview of Startup-configuration via Startup-config object

The EVE-NG includes the Export configuration feature, this feature allows saving the startup configuration of the nodes.
Note = When you import your lab into another EVE-NG server, this feature helps to keep saved the startup-config on nodes. You need just right-click on the node, then select the Export CFG, see in Figure 1. 9, and don't forget to keep the node's switch on inside the Startup-config object, see Figure 1. 8:

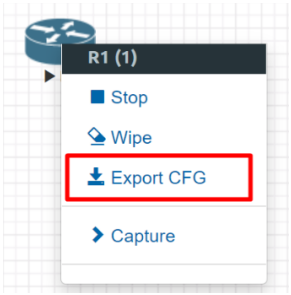


Figure 1. 9: EVE Export configuration feature

Configured objects: The Configured objects provides a list of the added images and shapes.
More actions: It provides mass control such as start all nodes, stop all nodes, consoles to all nodes, etc.
Refresh topology: In the case of having a mass topology of nodes, it needs sometimes to refresh the lab.
Status: The Status object provides an overview of the resource utilization such as CPU, RAM, disk utilization, and information like the number of nodes in a lab. The following Figure 1. 10 shows the lab status for the designed topology in Figure 1. 7:

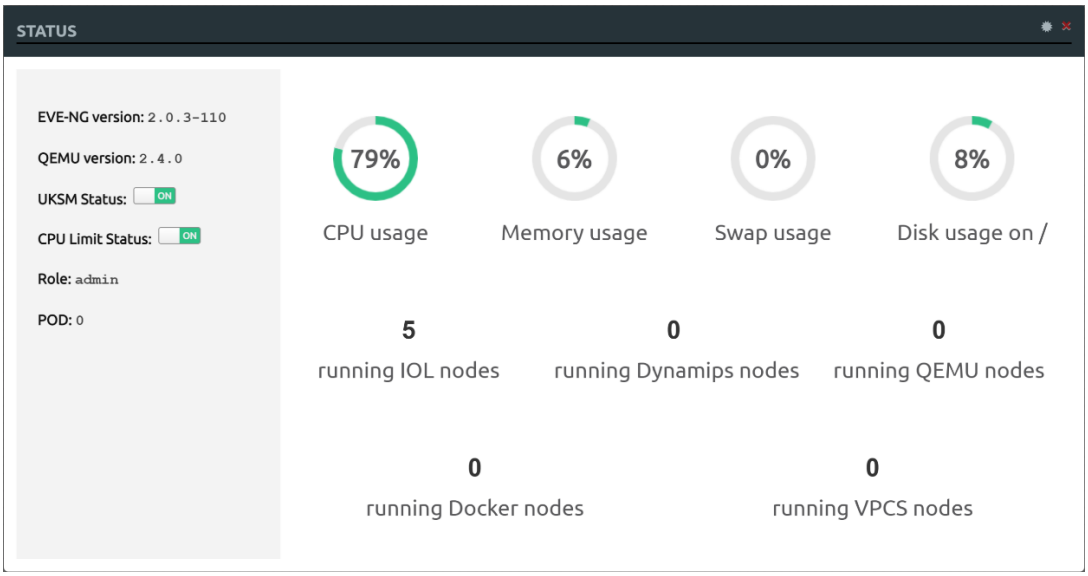


Figure 1. 10: EVE Status window

Lab details: The Lab details object provides information about the lab tasks and the lab's UUID:

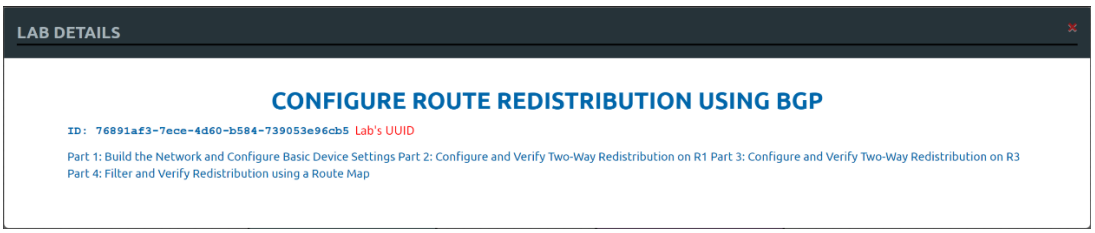


Figure 1. 11: EVE LAB details

Lock lab: The Lock lab object makes the lab mode "read-only" except for the lab settings.
Dark Mode: The Dark Mode object sets the lab background in two modes of dark or light.
Close lab: The Close lab object closes the lab.
Logout: The logout object provides to log out from the EVE WEB GUI session.

1.3.3 Uploading the emulator images

Uploading the emulator images inside the eve-ng server requires login to the server (via management IP address) by using FTP client tools such as WinSCP or FileZilla.

Each emulator must be uploaded in its dedicated directory:

- **Dynamips** - /opt/unetlab/addons/dynamips
- **Qemu** - /opt/unetlab/addons/qemu
- **IOL/IOS** - /opt/unetlab/addons/iol/bin/

We deployed the laboratories on both emulation environments but found, there are only two ways to get access to prepared labs remotely in GNS3: **1)** Installing GNS3 on a remote server that uses the OpenVPN software, **2)** Using GNS3 Web-UI with a separately designed VPN solution. The EVE-NG environment has been selected as the most suitable environment, as GNS3 Web-UI (BETA version) is buggy [46] and less user-friendly, and has an unstable connection with the GNS3 installation on the remote server that uses the OpenVPN software. The following chapter explains how to prepare the EVE-NG with the emulators for emulating laboratory tasks.

2. SETTING UP THE EVE-NG

A Client-Server model is designed for the remote access solution and the EVE-NG ISO file is installed at the server-side on physical hardware. All the required emulator images are found on the Internet, the following two emulators have been used:

- **Qemu**: Windows 7_32-bits is used in this thesis as a Qemu.
  Windows 7 has been installed with a terminal emulation program (Putty), Kiwi Syslog Server software, TFTP Server software, Power SNMP, and Elektron (Radius Server)

- **IOL/IOU**:
  The following three images L2/3 have been used for the Laboratory tasks:

  - L3-ADVENTERPRISEK9-M-15.4-2T.bin
  - L2-ADVENTERPRISEK9-M-15.2-20150703.bin
  - i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin

2.1 How to create a custom windows host (QEMU) for EVE

In the beginning, it needs to download a Windows ISO file format. The following ISO file has been used ((GetMyOS)Windows_7_Ultimate_X86_SP1_En_Aug_2018). The ISO image must be uploaded in the following directory: **/opt/unetlab/addons/qemu/**

- Login to the EVE-NG server, Create a new directory for this image.
  Note = New directory naming starts after "win- "as per EVE-NG documentation. When EVE-NG is looking for the images, it looks for folders with a specific name.
  Note = To get into the EVE-NG server, use SSH clients such as Putty, SecureCRT.

```
root@eve-ng:~#mkdir /opt/unetlab/addons/qemu/win-7-ENARSI/
```

- Upload the image file to the newly-created directory. The ideal way to perform this task is by using FTP clients such as WinSCP or FileZilla, see Figure 2. 1.
  Note = Host: 192.168.150.150, Username: root, Password: eve, Port #: 22.

  Confirm the image is uploaded, using the following command:

```
root@eve-ng:/opt/unetlab/addons/qemu# ls -l
drwxr-xr-x 2 root root 4096 Jan 5 18:35 win-7-ENARSI
```

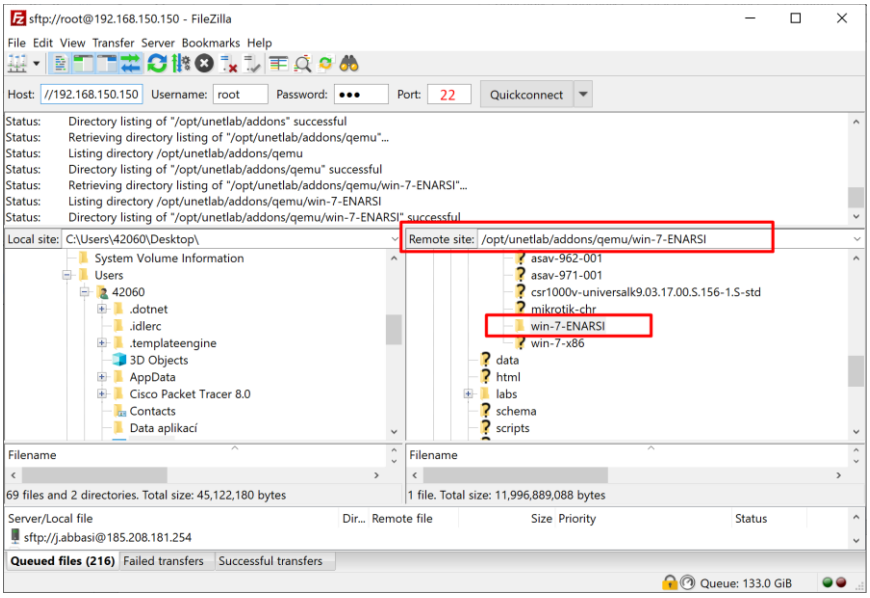


Figure 2. 1: How to Copy Windows ISO file format by FileZilla software

- The ISO file must be renamed to cdrom.iso.
  Move to the created directory (win-7-ENARSI), using the following command:

```
root@eve-ng: cd /opt/unetlab/addons/qemu/win-7-ENARSI
```

  Rename the file using the mv command:

```
Mv \(GetMyOS\)Windows_7_Ultimate_X86_SP1_En_Aug_2018.iso
cdrom.iso

root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI# ls
cdrom.iso
```

- Create a new virtual hard disk named virtioa.qcow2.
  Note = Inside the image folder must be placed HDD image with the correct name format, like the virtioa for Windows host (Qemu emulator). The ".qcow2" is a storage format for the virtual disk.
  Note = It is selected 50G for the HDD size.

```
root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI#
/opt/qemu/bin/qemu-img create -f qcow2 virtioa.qcow2 50G
```

- Browse the server IP address (192.168.150.150) in the browser. Use admin/eve as the Username/Pass to log in to the EVE-NG environment. Create a new Lab and add a new win-7-ENARSI host. The new host should be connected to the internet (Home LAN) via the Management Cloud node built-in EVE-NG. Start the win node and install Windows 7, see Figure 2. 2.

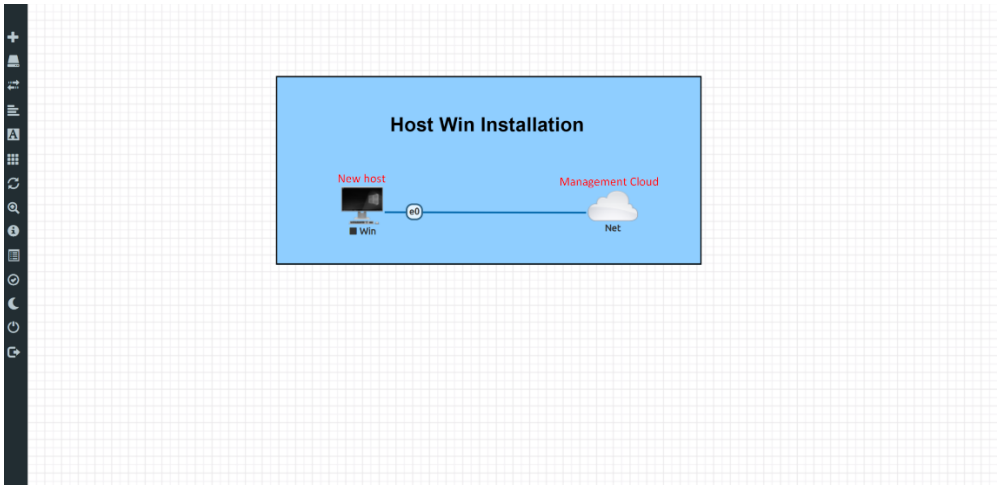


Figure 2. 2: Connecting new host to the internet via the Management Cloud

The following path directory must be selected when the installation asks to choose a driver where do you want to install Windows":

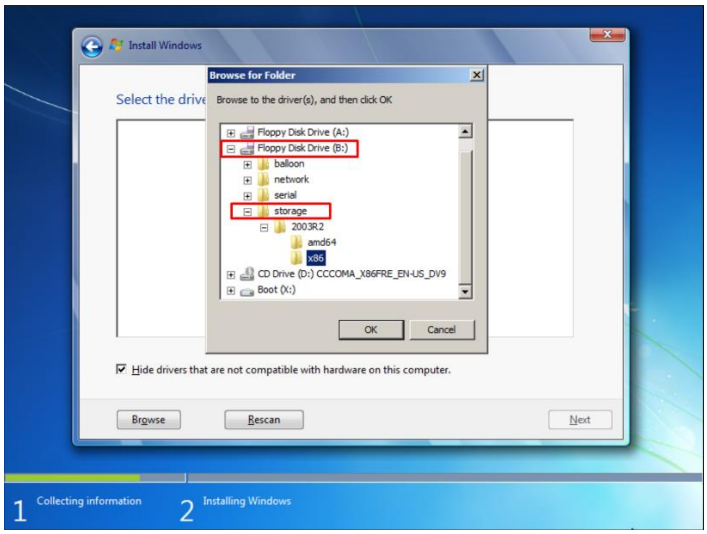
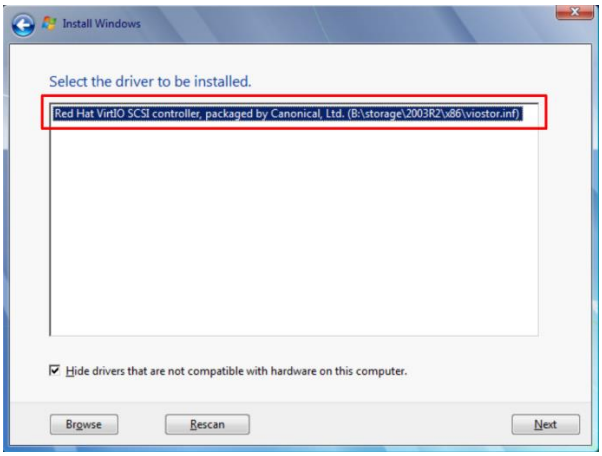


Figure 2. 3: Select the directory where the windows will be installed

The following HDD (RedHat VIRTIO SCSI) must be selected as the driver:



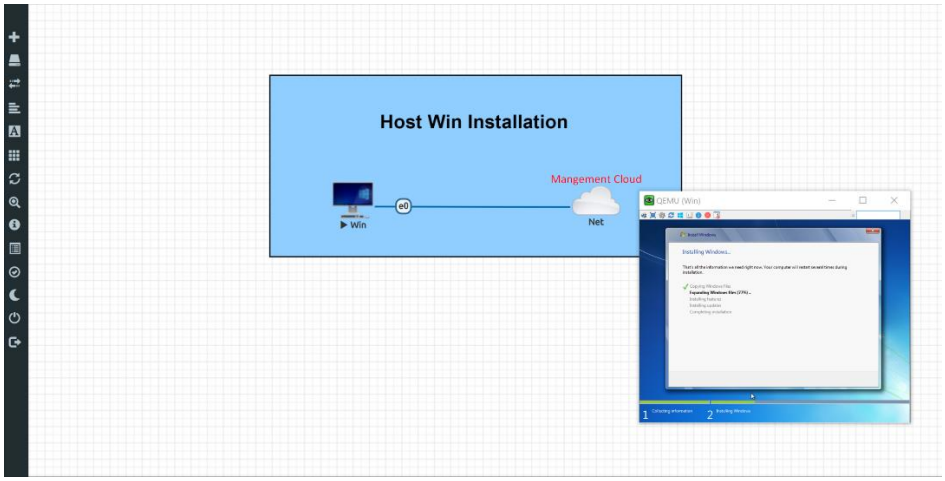


Figure 2. 4: Host Windows Installation

- Install the required Software for labs then commit the changes to set it as the default image for further use.
  Note = Terminal emulation program (Putty), Kiwi Syslog Server software, TFTP Server, and Elektron AAA Server software are installed for the laboratory tasks.

  Locate the installed image:

```
root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI#cd
/opt/unetlab/tmp/0/bb3a63ec-ef76-4a1b-826e-c03730271385/1/
```

  - **Lab ID**: bb3a63ec-ef76-4a1b-826e-c03730271385
  - **POD ID**: 0 (By default, it is assigned 0 for the admin users).
  - **Node ID**: 1

  The following figures verify the above IDs in the EVE-NG GUI:

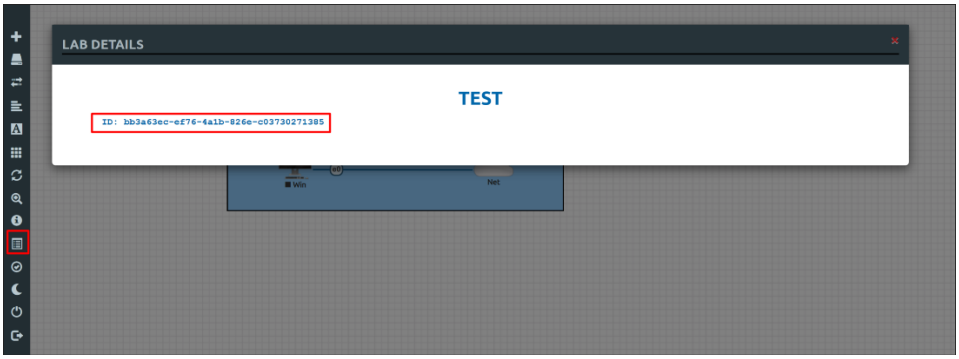


Figure 2. 5: Lab ID located in the “Lab Details”

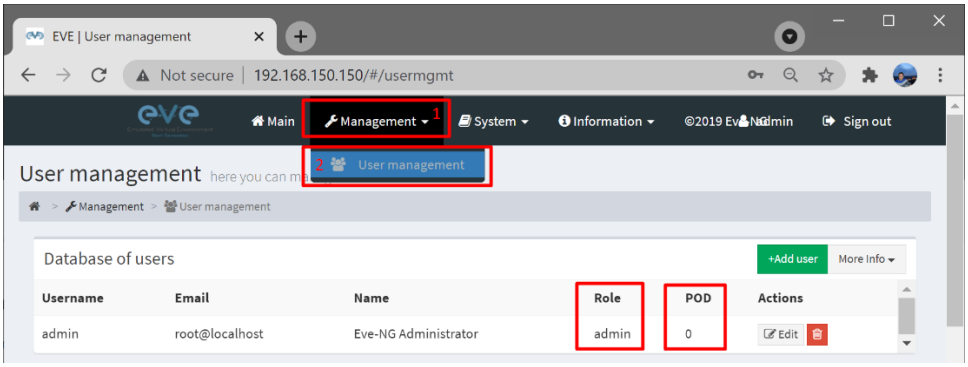


Figure 2. 6: POD ID in the EVE GUI under Management > User Management

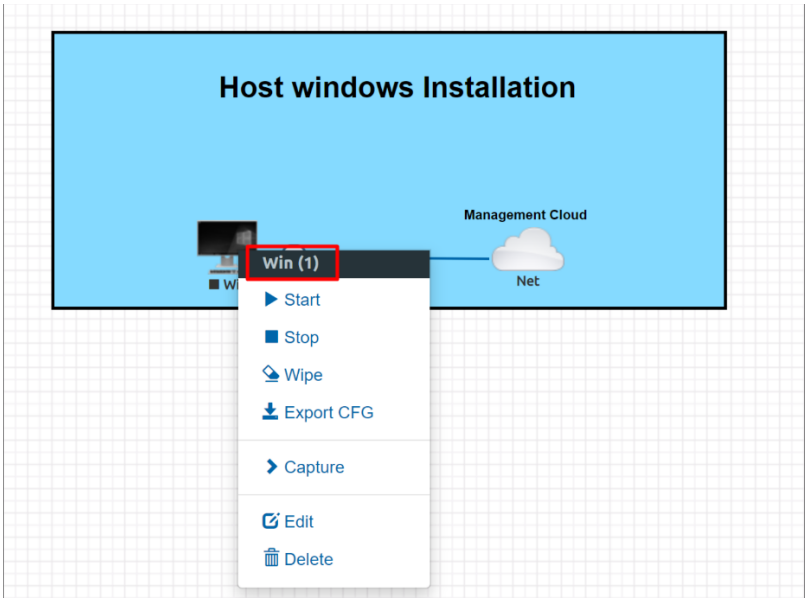


Figure 2. 7: Node ID is found by right-clicking the host node

Save the changes to the local repository (virtioa.qcow2):

```
root@eve-ng:/opt/unetlab/tmp/0/bb3a63ec-ef76-4a1b-826e-c03730271385/1#/opt/qemu/bin/qemu-img commit virtioa.qcow2
Image committed.
```

In the end, it is recommended to remove the cdrom.iso file:

```
root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI# ls
cdrom.iso  virtioa.qcow2

root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI# rm -f
cdrom.iso

root@eve-ng:/opt/unetlab/addons/qemu/win-7-ENARSI# ls
virtioa.qcow2
```

2.2 How to Upload IOL/IOU and generate a license key

To emulate laboratory tasks, various emulators such as Dynamips (Cisco IOS 3725 and Cisco IOS 7206VXR), Qemu (CISCO CSR 1000v (XE 3.x)) were tested but unable to support all the commands. Because of the lack of support, it has been decided to use the IOL. Different IOL images have been tested but the following three IOL images can support all the commands:

- L3-ADVENTERPRISEK9-M-15.4-2T.bin
- i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin
- L2-ADVENTERPRISEK9-M-15.2-20150703.bin (It is used for Switches in Lab: #23.1.4 Troubleshoot IP SLA and NetFlow).

The following steps explain how to upload the IOL images to an EVE-NG server and generate a license key:

- First of all, upload the images to the following directory **/opt/unetlab/addons/iol/bin/.** The ideal way to perform this task is by using FTP clients such as WinSCP or FileZilla.
  Note = Host: 192.168.150.150, Username: root, Password: eve, Port #: 22.

  Verify the images in the associated directory using the ls command:

```
root@eve-ng:~# ls /opt/unetlab/addons/iol/bin/
i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin
i86bi-linux-l3-adventerprisek9-15.5.2T.bin
L2-ADVENTERPRISEK9-M-15.2-20150703.bin
L2-adventerprisek9-ms.july3_2015_team_track_dsgs_pi5.bin
L3-ADVENTERPRISEK9-M-15.4-2T.bin
L3-ADVENTERPRISEK9-M-15.5-2T.bin
```

- Run the following command to set the permissions for the IOL/U images:

```
root@eve-ng:~#  /opt/unetlab/wrappers//unl_wrapper -a
fixpermissions
```

- Generate the iou license key under the EVE-NG to run the IOU/IOL image as per the following steps:

  - Move to the following directory:

```
root@eve-ng:~# cd /opt/unetlab/addons/iol/bin
```

  - Create a new file as Cisco iou keygen file:

```
root@eve-ng:/opt/unetlab/addons/iol/bin# sudo touch CiscoIOUKeygen.py
```

- Copy the following script to the file (Script is available on the internet):

```
root@eve-ng:/opt/unetlab/addons/iol/bin# sudo nano CiscoIOUKeygen.py
```

IOU license generator script:

The following script can be found on the internet as per the EVE-NG recommendation:

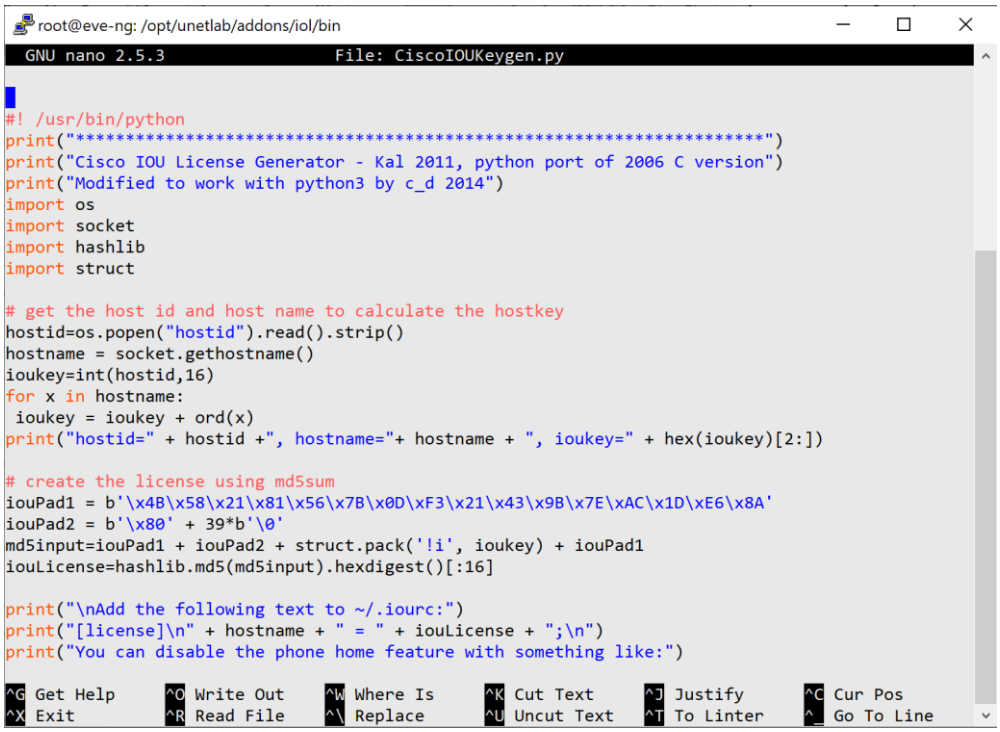


Figure 2. 8: IOU license generator script

- Make the script file executable using the following command:

```
root@eve-ng:/opt/unetlab/addons/iol/bin# sudo chmod +x CiscoIOUKeygen.py
root@eve-ng:/opt/unetlab/addons/iol/bin# ls -l
total 810880
-rwxr-xr-x 1 root root      1057 Apr 23 17:42 CiscoIOUKeygen.py
-rwxr-xr-x 1 root root 103040504 Jan  7 05:53 i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin
-rwxr-xr-x 1 root root 172982492 Jan  7 05:54 i86bi-linux-l3-adventerprisek9-15.5.2T.bin
```

- Execute the license generator script using the following command: #python3 CiscoIOUKeygen.py

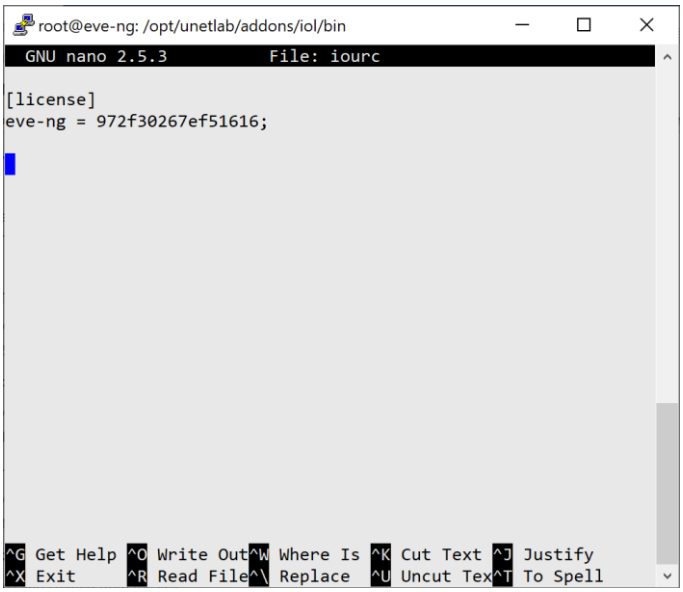
```
root@eve-ng:/opt/unetlab/addons/iol/bin# python3 CiscoIOUKeygen.py
********************************************************************
Cisco IOU License Generator - Kal 2011, python port of 2006 C version
Modified to work with python3 by c_d 2014
hostid=007f0101, hostname=eve-ng, ioukey=7f0343

Add the following text to ~/.iourc:
[license]
eve-ng = 972f30267ef51616;

You can disable the phone home feature with something like:
 echo '127.0.0.127 xml.cisco.com' >> /etc/hosts
```

- Create the iourc file to save the above-generated license key in it:

```
root@eve-ng:/opt/unetlab/addons/iol/bin# sudo touch iourc
root@eve-ng:/opt/unetlab/addons/iol/bin# sudo nano iourc
```




```
root@eve-ng:/opt/unetlab/addons/iol/bin# ls
CiscoIOUKeygen.py
i86bi-linux-l2-adventerprisek9-15.6.0.9S.bin
i86bi-linux-l3-adventerprisek9-15.5.2T.bin
iourc
```

- Set the permissions for the IOL/U images:

```
root@eve-ng:~#  /opt/unetlab/wrappers//unl_wrapper -a
fixpermissions
```

3. REMOTE ACCESS SOLUTION

The main part of the thesis is related to design and implement a remote access solution to prepared laboratory tasks. This problem was solved with a Virtual Private Network (VPN) solution. VPN provides an encrypted tunnel across a public network such as the internet, the end-user can get access to any remote network resources like a direct connection. In regards to a VPN service, MikroTik Layer 2 Tunneling Protocol (L2TP) Server has been selected. MikroTik L2TP Server provides an encrypted tunnel (only for control messages) across the internet for transporting data using Point-to-Point Protocol (PPP) [49]. Mikrotik L2TP Server can be implemented in the two following methods:

- **Site-to-Site L2TP:**
  This is a VPN service between two routers. In this method, a router (L2TP Client) creates a tunnel with another router at the remote site (L2TP Server).
- **Connecting remote workstation/client:**
  This is a VPN service between the L2TP-supported Operating System (OS) as an L2TP Client and the Mikrotik router as an L2TP Server. The Client can reach out to network resources by establishing an L2TP tunnel with L2TP Server. The goal is to connect remote clients to the EVE-NG server by using L2TP over IPsec (L2TP/IPsec) VPN Protocol.
  Note = L2TP protocol provides a layer 2 tunnel without any encryption of data, then it may be passed over Internet Protocol security or IPsec (layer 3 encryption protocol).

Remote Access Solution is based on the Client-Server model which consists of two parts: **Client-side** and **Server-side**:

3.1 Server-side

The server is located in Herat-Afghanistan where I come from. Because of the lack of devices and I also was curious to have a real-time solution across a long distance, it is decided to deploy the server in Afghanistan. The Server-side consists of an EVE-NG bare Installed on a Dell WorkStation and Mikrotik Router, see Figure 3. 1. Dell Workstation is working as an EVE-NG server that has a direct connection to the router R1. LAN IP addresses are assigned from a range of 192.168.150.0/24 and WAN IP addresses are assigned from a range of 103.126.5.252/30. As mentioned above, the remote access solution is about configuring L2TP Server on R1 and L2TP/IPsec VPN on the Home-PC at the client-side. L2TP/IPsec Server creates an L2TP tunnel with an assigned IP address 192.168.150.101 across the public network. On the Client side, Home-PC has been configured with L2TP/IPsec VPN which assigned an IP address (192.168.150.202 or

192.168.150.203) from the range of server LAN IP pool that gives access to the local network resources.

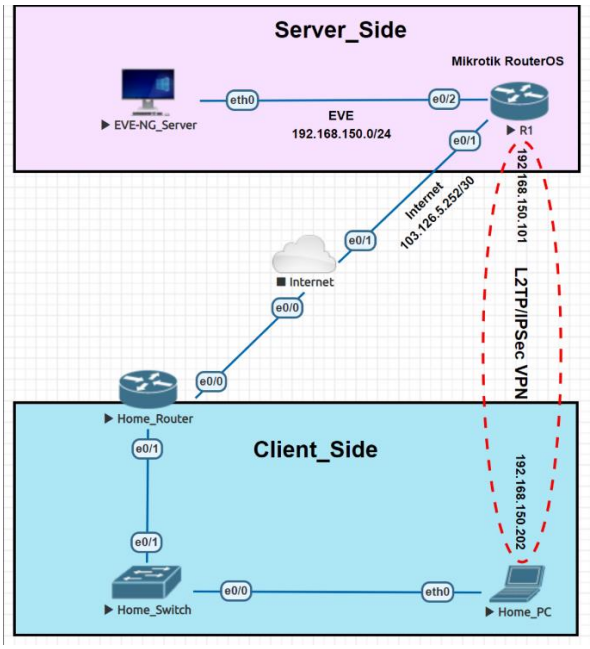


Figure 3. 1: Network topology for the remote access solution

3.1.1 Server-side configuration

The Server-side is configured in the five following steps:

- **Bare Hardware EVE-NG Server Installation**
  EVE-NG ISO image has been installed on Dell WorkStation as Bare Hardware EVE Server Installation.

- **Mikrotik RouterOS Basic Configuration**
  Router basic configuration such as assigning WAN and LAN IP addresses, Perform NAT, DNS, and route configuration should be considered in the first steps of configuration. Mikrotik Winbox is used to configure the router.
  Winbox: Winbox is a Mikrotik Administration utility using a simple GUI.

  Assign an IP address for the LAN interface, WAN interface, and DNS server:

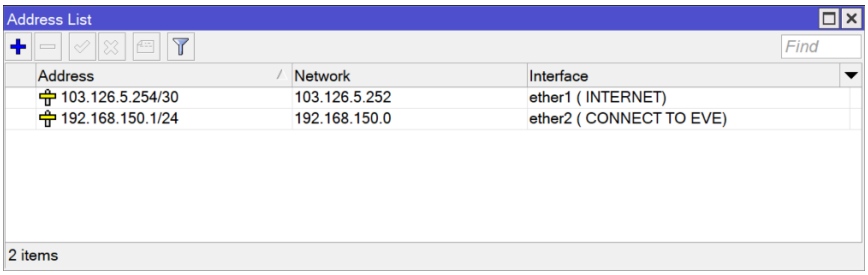


Figure 3. 2: Assigning the WAN and LAN IP addresses

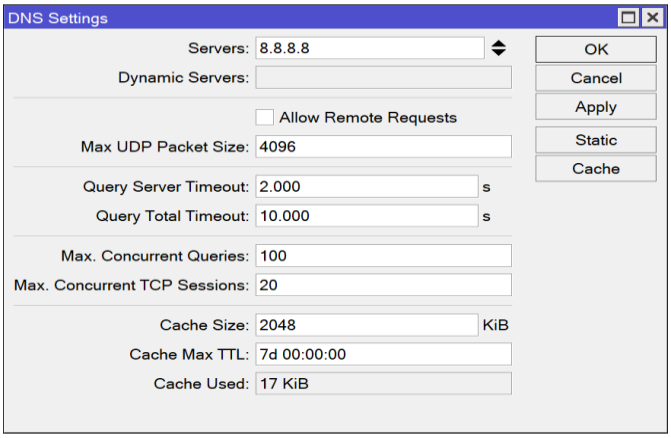


Figure 3. 3: Assigning DNS server IP address (8.8.8.8)

Configure the source network address translation (masquerading) feature. This feature hides our local devices behind the public address received, The IP address of the EVE-NG server (192.168.150.150) must be selected as the Src address:

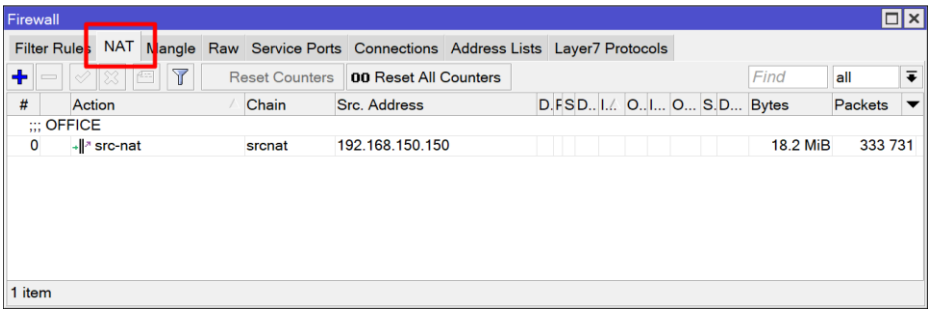


Figure 3. 4: Source NAT Configuration

Configure the default static route with the following IP address for Gateway 103.126.5.253:

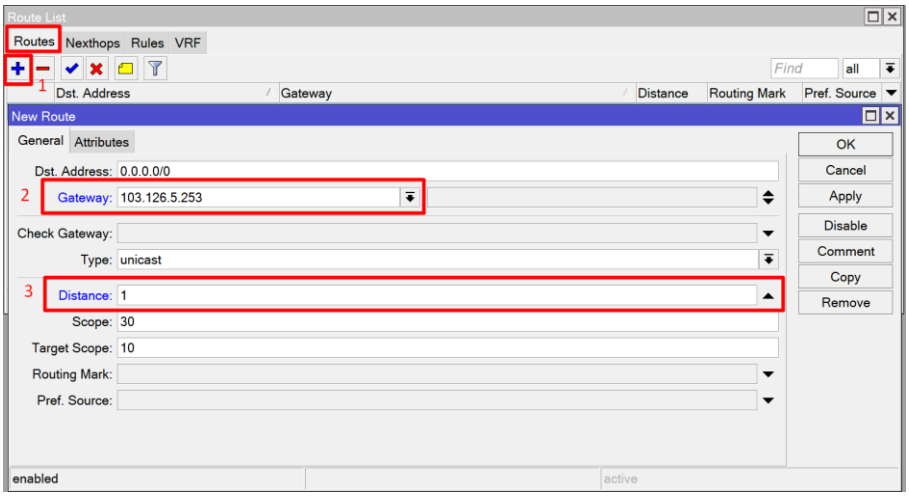


Figure 3. 5: Route Configuration

➢ **L2TP Server Configuration**

Enable L2TP server, Authentication, IPsec, and its password:

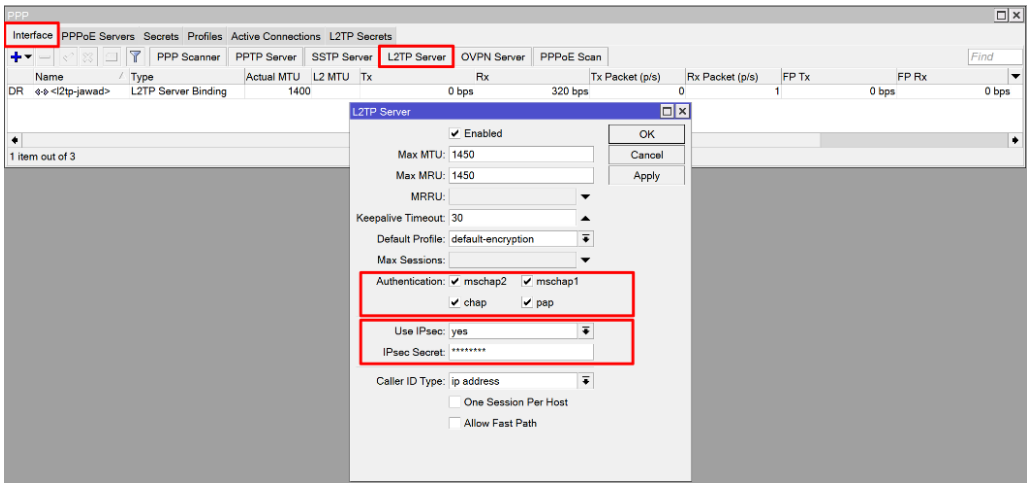


Figure 3. 6: L2TP Server Configuration

➢ **Creating PPP Secretes for L2TP Server**

Configure credentials (Username/Password) for the Client-side when access requests:

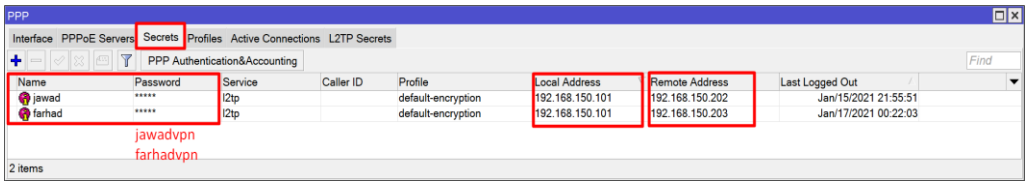


Figure 3. 7: Creating PPP credentials for L2TP server

➢ **Enabling Proxy-arp on LAN interface**

We have to enable proxy-arp for the LAN interface (EVE) to give reachability for Client-Side users.

Note = Proxy ARP answers the ARP queries for a network address that is not on that network that the request comes from. Router R1 works as a proxy and this is called proxy-arp so Proxy ARP is a service that can be running on a router.

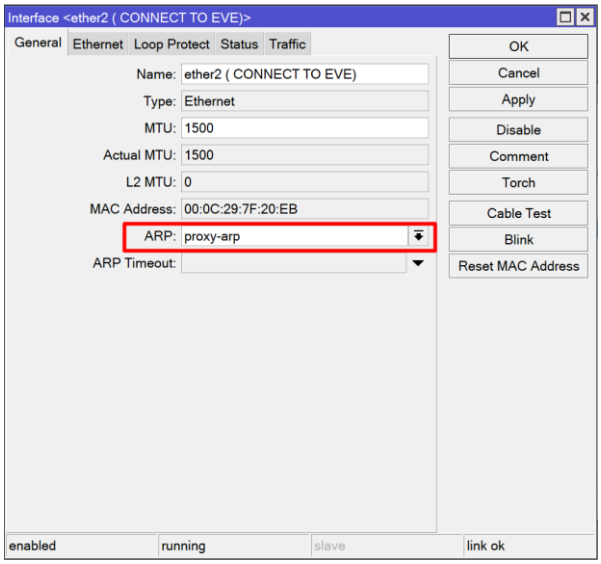


Figure 3. 8: Enabling Proxy-ARP on LAN interface

3.2 Client-side

The Client-side has been configured with L2TP/IPsec VPN protocol on Windows 10. L2TP/IPsec has been selected as the VPN type which is a combination of Layer 2 Tunneling Protocol and Internet Protocol Security that enables a secured connection over a public network. L2TP/IPsec is built-in on Windows, Mac, Android, and iOS. L2TP consists of two components: 1) Tunnel and 2) Session. The tunnel provides reliable transport to carry only control packets. The Session is logically located inside a Tunnel that carries only user data [50]. L2TP relies on IPsec because doesn't provide any data authentication and encryption mechanism.

3.2.1 Client-side configuration

The Client-side is configured in the eight following steps:

- Open Control Panel > Network and Internet > Network and Sharing Center. Select "Set up a new connection or network" to set up a VPN connection:
- Select "Connect to a workplace":
- Select "Use my Internet connection (VPN)":
- Insert the following public IP address of the VPN server into the "Internet Address" section: IP Address 103.126.5.254. Name the VPN as you wish.
- Open the created VPN properties, in the "Security" section select the type of VPN: Layer 2 Tunneling Protocol with IPsec (L2TP/IPsec)
- Go to "Advanced settings" and select the "pre-shared key" option for authentication (Pre-Shared Key: jawad!@#):
- In the "Networking" section > Internet Protocol version 4 (TCP/IPv4) > Properties > Advanced > Disable "Use default gateway on remote network", by disabling this option you are connected to the internet through your local gateway:

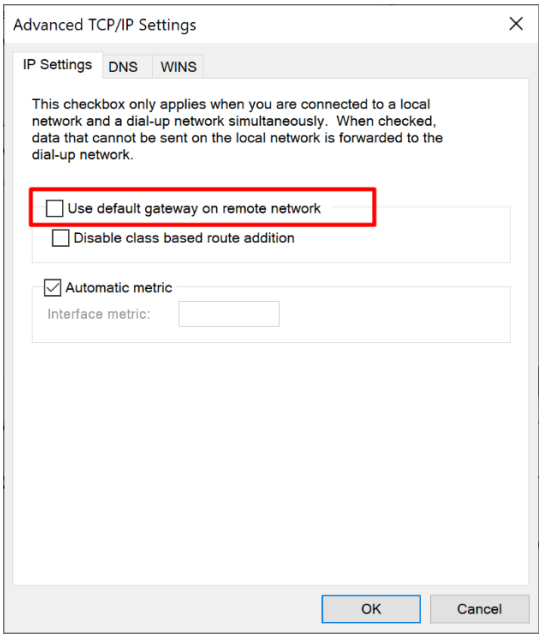


Figure 3. 9: Disabling "Use default gateway on remote network".

- When trying to get connect to the L2TP server (103.126.5.254), OS installed on PC asks for the credentials (name and password), it is configured with two credentials on the server-side, see Figure 3. 7:

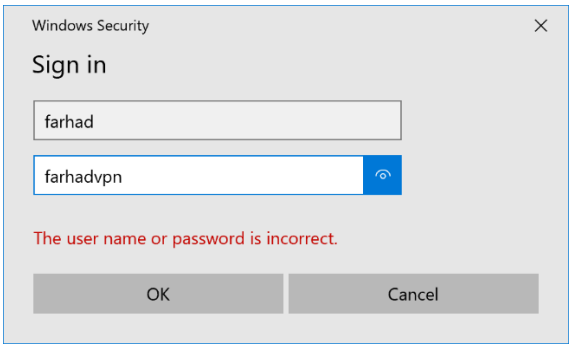


Figure 3. 10: The 8th step of VPN configuration on client-side

- Open a new tab on your browser and write the EVE management IP address (192.168.150.150). Use the default web login admin/eve.

The following chapters from 4 to 15 are related to the topics of the ENARSI certification exam. There are 40 laboratories provided, that should be emulated. These labs consist of two types of Implementation and Troubleshooting, each Troubleshooting lab contains 1 or more trouble tickets. All labs are emulated and available on the EVE-NG server. A topology is created for each laboratory, devices within Implementation labs are configured with the initial configurations, and a configuration file (.txt) is created and uploaded to the device in Troubleshooting labs. In the following chapters, some of the configuration commands are explained with the help of the emulated "Implementation Labs" on the EVE-NG server, and seven trouble tickets have been worked out and explained for different topics.

4. IPv4/IPv6 Addressing and Static Routes

This chapter covers the two fundamental topics in networking which are IP addressing and Static routes. The following labs are dedicated to the topics of this chapter:

- Lab **#1.1.2**_Troubleshoot IPv4 and IPv6 Addressing Issues (It contains 3 trouble tickets 1.1.2.1-3).
- Lab **#1.1.3**_Troubleshoot IPv4 and IPv6 Static Routing (contains 2 trouble tickets).

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

4.1 IPv4 Addressing

An IP address is a unique logical address assigned to a network device in an IP network. The IP address is separated into two parts by a subnet mask: A) Network portion, B) Host portion, the Host portion should be different on each device in the same net/subnet. An IPv4 consists of a 32-bits length which is divided into 4 octet blocks (bytes). The dotted-decimal format is used to read an IPv4 address [3].
Sometimes there is a network problem with IP addressing, which can be due to a mistake either misunderstanding of IP addressing. To troubleshoot these problems as soon as possible, it is recommended to consider the following points [1]:

- Improper IP address
- Unsuitable subnet mask
- Default gateway configuration with the wrong IP address

The IP address of a Windows host is verified by the command ipconfig [1] and the show ip interface *interface_type interface_number* command verifies the IP address of an interface on an IOS Router or Switch [1].

4.1.1 DHCP for IPv4

Dynamic Host Configuration Protocol (DHCP) is an application protocol for dynamically assigning an IP address to any device on the network [3]. It works based on a Client-Server model. DHCP-Server not only assigns an IP address to DHCP-Client but also manages the network configuration for the subnet mask, default gateway IP address, DNS service. DHCP-Server can be implemented within the same network or inside a remote network. DHCP for IPv4 uses UDP port numbers 67 (Client) and 68 (Server) [3],[1].

To dynamically assign an IP address, DHCP Server and Client must go through the DORA process. The DORA process is about exchanging messages (Discovery, Offer, Request, and ACK) between the server and the client [1].

The following Wireshark capture in Figure 4. 1 verifies the DORA process on the emulated network in Figure 4. 2 between PC1 as a DHCP client and R2 as a DHCP server:

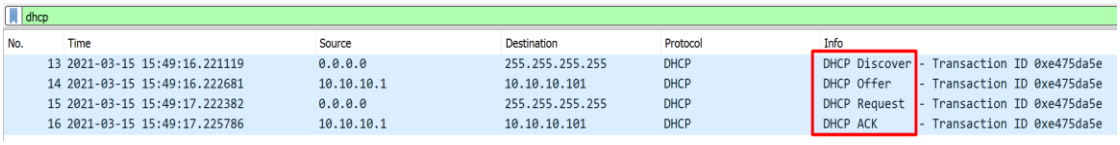


Figure 4. 1: Verify the DORA process in Wireshark.

- **Discovery**: The client starts the Dora process by sending a Discover message, the message is sent to broadcast address 255.255.255.255 with the source address "0.0.0.0" [1], [3].
- **Offer**: The DHCP server responds to the Discover message with the Offer message which offers the IP addressing information. Because the client sent a broadcast message, probably it receives more than one Offer message from different servers. The client chooses the unleased address in the first Offer message[1], [3].
- **Request**: The DHCP client notifies the server to use the IP address provided in the previous message with the "Request" message [1], [3].
- **Acknowledgment**: Finally, the server approves the client request with an "Ack" message. This confirms that the IP has been leased to the client [1], [3].

In real network operation, the DHCP server is mostly not located within the local network. The local gateway must be configured as the "DHCP relay agent" with the interface configuration mode command ip helper address *ip_address*. This is because the "Discover" broadcast message cannot pass through the router to reach the DHCP server on the remote site. It's worth mentioning that the DHCP server can also be configured on the same network/subnet default gateway [28], [1].

4.2 IPv6 Addressing

IPv6 was originally defined in RFC 2460 – Internet Protocol, Version 6 (IPv6) Specification [29]. The main difference with IPv4 is the amount of address space, IPv6 consists of 128 bits in length which provides 2^{128} ~ 340 undecillion (3.4×10^{38}) addresses [3]. In terms of addressing format, IPv6 uses Hexadecimal formatting [3]. In general, IPv6 addresses are divided into 3 types of addresses: Unicast, Multicast, and Anycast (defined in RFC 4291 – IP Version 6 Addressing Architecture) [3]:

- **Unicast**: Each network interface is identified by a unicast address for exchanging packets.
- **Multicast**: Packets are sent to a group of hosts, which also represents the broadcast address that is not defined in IPv6.
- **Anycast**: The recipient of this address is only one member of the group, while the entire group is the recipient in the Multicast address

Regarding the division of IPv6 address types, Unicast address is also divided into the following three types [3]:

- **Link-Local** (FE80::/10_equivalent of 169.254.0.0/16 in IPv4): This type of address is locally significant to a link which makes them never routable. It is used for Router Discovery, Neighbor Discovery and Stateless Address Auto-configuration (SLAAC) [27].
- **Unique Local** (FC00::/7_equivalent of Private address in IPv4): This type of address is likely unique inside an organization that is not routable globally [27].
- **Global Unicast** (2000::/3_quivalent of Public address in IPv4): This type of address is a publicly routable address [27].

An IPv6 address is divided into two parts. The first portion is called Prefix (first 64 bits) and the second portion is called Interface/host ID (64bits). It is recommended to assign the Interface ID automatically with random numbers (it is used in Windows PC) or using the EUI-64 standard. The EUI-64 standard takes the Mac address of a link/host then separates the 48 bits length of the Mac Address into two 24 bits. Then inserts hex values of 0xFF 0xFE in the middle and changes the 7th bit from (0 to 1) or (1 to 0). By default, Cisco routers use the EUI-64 standard to assign the Interface ID of the link-local address of an interface but to enable the EUI-64 standard for the Interface ID of a global address on an interface, it requires to insert the "eui-64" keyword at the end of the ipv6 address command [1].
An IPv6 uses the following 4 types of messages regarding neighbor discovery [27]: **1) Neighbor Solicitation (NS)** which is equivalent of ARP request in IPv4, **2) Neighbor Advertisement (NA)** which is equivalent of ARP reply, **3) Router Solicitation (RS)_** to find the default gateway, and **4) Router Advertisement (RA)** which is sent by a gateway in response of RS.

4.2.1 DHCP for IPv6

There are three methods to lease an IPv6 address dynamically: 1) Stateless Address Autoconfiguration (SLAAC), 2) Stateful DHCPv6, 3) Stateless DHCPV6. DHCP for IPv6 uses UDP port numbers 546 and 547 [1]:

- **SLAAC**: Stateless Address Autoconfiguration or SLAAC (defined in RFC 4862) is a feature that allows each device in the same subnet to configure its IPv6

address, Prefix, and Default gateway address. To enable SLAAC on Cisco interfaces, the interface should be configured with the ipv6 address autoconfig command. The SLAAC feature works based on two messages of RS and RA. Each interface enabled with the SLAAC feature or Windows PC (enabled by default) sent out the RS message to find the default gateway connected to the local link, and in response to RS, the gateway replies with the RA message providing the Prefix information. In addition, router interfaces use the EUI-64 standard to create the Interface/host ID of the IPv6 address. Once the host gets the address, uses Duplicate Address Detection (DAD) function to confirm the uniqueness of the generated address. In this method, the default gateway is identified with its link-local address, and other parameters such as DNS server information are not provided [1], [27].

- **Stateful DHCPv6**: Nowadays, devices in modern enterprise networks require not only basic prefix information but should be provided other necessary information such as Domain Name System (DNS) server address, Network Time Protocol (NTP) server address, etc. This all can be obtained by a separate DHCPv6 server or enabling DHCPv6 server function on the Cisco routers and Multilayer switch. To inform an interface to use DHCP pool on DHCPv6 server, it must be configured with ipv6 DHCP server interface command [1], [27].
- **Stateless DHCPv6**: This method is the combination of two methods of SLAAC and Stateful DHCPv6 to get the prefix information and other information that can be only provided by Stateful DHCPv6 [1].

An IPv6 address gets assign dynamically if the DHCPv6 Server and Client should pass through the DHCPv6 operation [1]:

- **Solicit** = The client initiates the DHCPv6 operation by sending the Solicit message. It using all DHCPv6 server's multicast address FF02::1:2.
- **Advertise** = The DHCPv6 server responds to Solicit message with a unicast Advertise message which offers the addressing information.
- **Request** = The client in response to the Advertised message sends this message to request the information.
- **Reply** = Finally, the server acknowledges the client request with a "Reply" message. This will confirm that IP is leased to the client.

4.3 Static Routes

Static Routing is a type of routing in computer networks in which the data path is already defined by the network administrator and the router does not need to process and only executes commands. Unlike dynamic routing, static routing remains unchanged until

there is a change in the physical structure of the network (such as adding or removing a router from the network) [1], [2]. The static route has an AD of 1 by default [1].

4.3.1 IPv4 Static Routes

The IPv4 static route will be established with the following global configuration command [1]:

```
ip route prefix mask {ip_address | interface_type interface_number}
[distance]
```

Syntax *ip-address* specifies the next-hop IP address that can be used to reach that network, this is while *interface_type* specifies the exit interface on the local device that can be used to reach the network [1].

4.3.2 IPv6 Static Routes

The IPv6 static route will be established with the following global configuration command [1]:

```
ipv6 route {ipv6_prefix/prefix_length} {ipv6_address | interface_type
interface_number} [administrative_distance] [next-hop_address]
```

4.4 Troubleshooting

This section covers problems you may encounter with IP addressing and static routes. In addition, troubleshooting techniques to solve these problems. Trouble ticket #1.1.2.2 from "Troubleshoot labs" has been worked at the end of this section, on behalf of other trouble tickets.

The following include Cisco recommendations for troubleshooting DHCP problems [1]:

- The DHCP relay agent is not configured on its interface that receives the "Discover" message from the DHCP client.
- Layer 2 problem, such as VLANs, Spanning Tree Protocol (STP), trunks.
- Configuring DHCP server with improper parameters.
- DHCP pool at the server is out of IP addresses.
- The DHCP server lost its connection with the redundant server, this may occur overlapping IP addresses issue.

IPv4 Static route: It is recommended to establish a static route with specifying the exit interface in a point-to-point link such as serial or DSL [1].
IPv6 Static route: When the link-local address is used as the next-hop address, the exit interface also should be specified [1]. This is because of the probability of having the same link-local address on different remote/local router interfaces.

4.4.1 Ticket #1.1.2.2 (Troubleshoot IPv4 and IPv6 addressing)

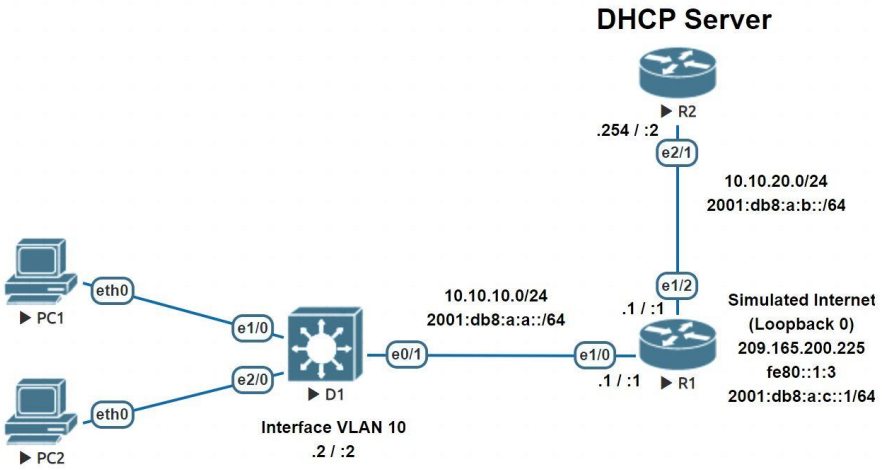


Figure 4. 2: Network topology for the emulated lab "Troubleshoot IPv4 and IPv6 addressing"

In this trouble ticket, both PC1 and PC2 are unable to lease an IPv4 address from the DHCP server. The DHCP server is located out of their network, configured on R2:

```
PC1> ip dhcp
DDD
Can't find dhcp server
```

This problem could be due to layer 1 (connectivity), layer 2 (VLANs, STP, trunks, etc), incorrect parameter settings on the DHCP server, or DHCP relay. We assume that there are no Layer 1 or 2 issues in this network. Let’s move on to R2 to check the configured parameters on the DHCP server. The output of the following command in Figure 4. 3 verifies that R2 is configured as a DHCP server with a DHCP pool but it was found that no DHCP message was received on the server:

```
R2#sh ip dhcp server statistics
Memory usage         24168
Address pools        1
Database agents      0
Automatic bindings   0
Manual bindings      0
Expired bindings     0
Malformed messages   0
Secure arp entries   0

Message              Received
BOOTREQUEST          0
DHCPDISCOVER         0
DHCPREQUEST          0
DHCPDECLINE          0
DHCPRELEASE          0
DHCPINFORM           0

Message              Sent
BOOTREPLY            0
DHCPOFFER            0
DHCPACK              0
DHCPNAK              0
R2#
```

Figure 4. 3: Display DHCP server statistics using the show ip dhcp server statistics command.

There may be a problem with the DHCP pool, the lack of IP address or incorrect IP address range can count as possible DHCP pool problems. The following command in Figure 4. 4 verifies the DHCP pool with its configured parameters:

```
R2#show ip dhcp pool

Pool LAN4_10 :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 0
 Pending event                  : none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased addresses
 10.10.10.1           10.10.10.1       - 10.10.10.254      0
```

Figure 4. 4: Display information about the DHCP address pools using the show ip dhcp pool command.

The DHCP pool is configured with a valid range of 254 IP addresses. We can move back to check the DHCP relay, it can be a problem with improper interface configuration that receives the “Discover” message from the DHCP client. The following output verifies that the DHCP relay is configured with the wrong interface for the Helper address:

```
R1#show ip int e1/0
Ethernet1/0 is up, line protocol is up
  Internet address is 10.10.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by configuration file
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
```

```
R1#show ip int e1/2
Ethernet1/2 is up, line protocol is up
  Internet address is 10.10.20.1/24
  Broadcast address is 255.255.255.255
  Address determined by configuration file
  MTU is 1500 bytes
  Helper address is 10.10.20.254
  Directed broadcast forwarding is disabled
```

The wrong interface must be removed first, and then the helper-address must be configured for the correct interface:

```
R1(config)#int e1/2
R1(config-if)#no ip helper-address 10.10.20.254
R1(config-if)#exit
R1(config)#int e1/0
R1(config-if)#ip helper-address 10.10.20.254
R1(config-if)#end
```

PC1 leased an IP address from the DHCP server:

```
PC1> ip dhcp
DDORA IP 10.10.10.101/24 GW 10.10.10.1
```

5. EIGRP

This chapter covers in more detail the laboratory material related to the EIGRP routing protocol. The following labs are dedicated to the topics of this chapter:

- Lab **#2.1.2_** Implement EIGRP for IPv4
- Lab **#3.1.2_** Implement Advanced EIGRP for IPv4 Features
- Lab **#4.1.2_**Troubleshoot EIGRP for IPv4 (It contains 3 trouble tickets 4.1.2.1-3)
- Lab **#5.1.2_**Implement EIGRP for IPv6
- Lab **#5.1.3_** Troubleshoot EIGRP for IPv6 (It contains 1 trouble tickets 5.1.2.1)

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

Enhanced Interior Gateway Routing Protocol (EIGRP) is a Cisco proprietary protocol with a combination of the advantages of Distance Vector and Link State routing protocols. However, EIGRP is generally considered as an Advanced Distance Vector routing protocol that supports the following advanced feature compared to other Distance Vector routing protocols: 255 hops away, Variable Length Subnet Mask (VLSM), equal and unequal cost load balancing, EIGRP uses the Diffusing Update Algorithm (DUAL) to make the network convergence faster, it can be provided by precalculated loop-free backup paths. EIGRP has an Administrative Distance (AD) of 90 for internal EIGRP and AD of 170 for external EIGRP [1], [5].

To better understand the features of EIGRP and how it works, it is important to know the basic concepts of EIGRP, see Table 5. 1 [1]:

Term	Definition
Successor route	The route with the lowest path metric to reach a destination.
Successor	The first next-hop router for the successor route.
Feasible Distance (FD)	The metric value for the lowest-metric path to reach a destination. The feasible distance is calculated locally.
Reported Distance (RD)	The distance reported by a router to reach a prefix. The reported distance value is the feasible distance for the advertising router.
Feasibility condition	A condition under which, for a route to be considered a backup route, the reported distance received for that route must be less the feasible distance calculated locally. This logic guarantees a loop-free path.
Feasible successor	A route that satisfies the feasibility conditioned and is maintained as a backup route. The feasibility condition ensures that the backup route is loop-free.

Table 5. 1: EIGRP Terminology

5.1 EIGRP Tables

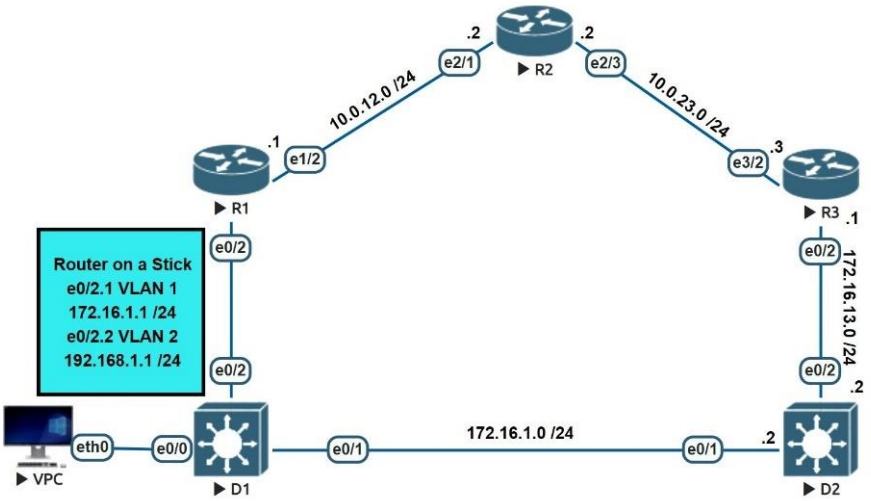


Figure 5. 1: Network topology for the emulated lab #2.1.2 " Implement EIGRP configuration for IPv4"

The EIGRP protocol has three following tables:

- **Routing table**: This table stores the best routes.
- **Neighbor table**: In this table, new adjacent neighbors are listed with their address and other necessary information such as Hold time, Uptime, Seq num, etc. [36]. The following Figure 5. 2 verifies the EIGRP neighbor table on R1 according to the emulated network in Figure 5. 1:

```
R1#show ip eigrp neighbors
EIGRP-IPv4 VR(BASIC-EIGRP-LAB) Address-Family Neighbors for AS(27)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   10.0.12.2               Et1/2                    12 00:00:58 1023  5000  0  4
```

Figure 5. 2: EIGRP Neighbor table

EIGRP has its own IP protocol number (88) and supports both multicast and unicast packets [1]. To initiate communication with other routers uses the multicast address of 224.0.0.10 and the related Mac address of 01:00:5e:00:00:0a [1]. A full exchange of routing tables between EIGRP neighbors happens during forming an adjacency. After full adjacency, only the incremental updates are advertised when a change occurs in topology [2]. The following Table 5. 2 is the list of 5 packet types that EIGRP uses to communicate with other routers [1]:

Packet Name	Function
Hello	This packet is used in two ways: Detecting the neighbors which are not in use, and another way when an EIGRP router received the Hello packets, tries to form an adjacency. The following parameters must be matched between them: a) Autonomous

	System Number (ASN), b) Primary Subnet, c) Authentication parameters, d) K values,
Request	Request to get information from other neighbors.
Update	Send routing table information to other EIGRP neighbors.
Query	Request for a specific path during convergence.
Reply	Reply to a specific route request.

Table 5. 2: EIGRP message types

- Topology table: This table contains the list of network prefixes that EIGRP has learned, their metrics, and the values used to calculate the metric [2]. The following Figure 5. 3 verifies the EIGRP topology table on R2 according to the emulated network in Figure 5. 1:

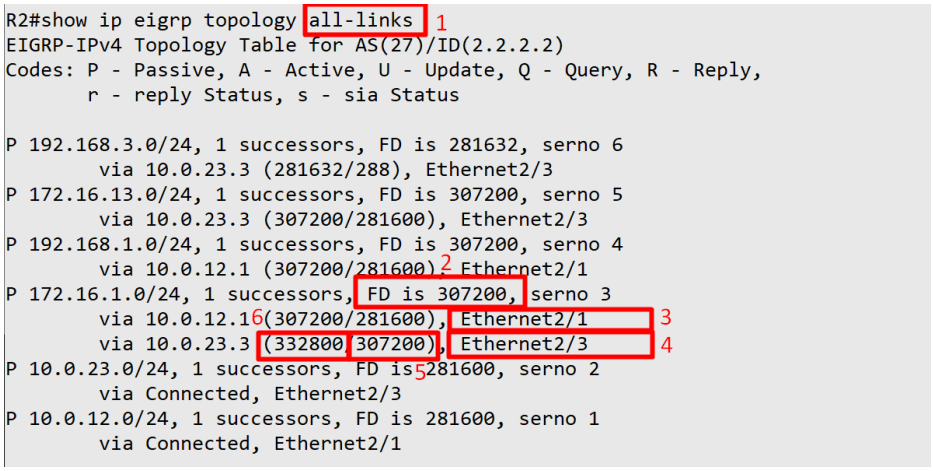


Figure 5. 3: EIGRP Topology table

- **1** (show ip eigrp topology all-links): show the full topology database.
- **2** (307200): Feasible Distance.
- **3** (Ethernet2/1): Successor route.
- **4** (Ethernet2/3): Didn't pass the Feasibility Condition RD (307200) = FD (307200).
- **5** (307200): Reported Distance.
- **6** (332800): Path Metric.

5.2 EIGRP Metrics

EIGRP supports two sets of metrics [1]:

5.2.1 Classic Metric

EIGRP uses different factors such as bandwidth (BW), delay, interface load, and reliability to calculate the metric [2]. In Classic metric, the router calculates the metric according to the following equation:

$$\boldsymbol{Metric = 256 * \left[\left(K_1 * \frac{10^7}{Min.BW} + \frac{\frac{10^7}{K_2 * Min.BW}}{256 - Load} + \frac{K_3 * Total\ Delay}{10}\right) * \frac{K_5}{K_4 + Reliability}\right]} \quad (5.1)$$

By default, K1=1 and K3=1, but the rest of the K values are equals to 0. BW represents the slowest link in the path scaled to a 10 Gbps link (10^7). Delay is the total measure of the delay in the path, measured in tens of microseconds (μs) [1].

$$\boldsymbol{Metric = 256 * \left(\frac{10^7}{Min.BW} + \frac{Total\ Delay}{10}\right)} \quad (5.2)$$

In the classic metric, the BW is in kilobytes per second and the delay is measured in 10 microseconds which makes scalability issue with the high-speed links. The classic metric is not able to calculate the exact metric of the links above 1G because has a constant value for the Bandwidth attribute also by decreasing the delay on each interface, the metric calculation does not have an exact output [1], [5].

5.2.2 Wide Metric

Due to increased link speed and reduced delay, it is recommended to use Wide Metric. In the Wide metric, the term Throughput is replaced by BW, Latency with delay, in addition, the value of K6 (K6 = 0) has been added to measure energy, jitter, and other future attributes [1], [5]:

$$\boldsymbol{Wide\ Metric = 65535 * \left[\left(\frac{K_1 * 10^7}{Min.BW} + \frac{\frac{K_2 * 10^7}{Min.BW}}{256 - Load} + \frac{K_3 * Latency}{10^{-6}} + K_6 * Extended\right) * \frac{K_5}{K_4 + Reliability}\right]} \quad (5.3)$$

5.3 EIGRP Configuration

EIGRP configurations can be implemented in two following modes:

5.3.1 Classic mode

Classic EIGRP mode configuration takes place in EIGRP process global configuration mode and partially in the interface configuration sub-mode [1]:

Initialization of the routing process:

```
router eigrp as-number
```

Enabling interface under the same prefix for EIGRP process:

```
network ip-address [mask]
eigrp router-id   Each router has a unique 32-bit router-ID
```

5.3.2 Name mode

There are some restrictions with classic mode e.g., it is required to be configured separate EIGRP processes for different EIGRP instances. EIGRP Name mode is a new method of the configuration since IOS version 15 is released, in which only a single EIGRP instance is required to be created, supports all Address Family types but must be activated for IPv4 with 'no shut', it supports multiple VRFs, and overcome the problem scattered configurations by configuring in one location [1], [30].

EIGRP name mode configuration stores the settings in three subsections [1], [30]:

- **Address Family**: Stores the configurations relevant to the global EIGRP AS operation such as networks, EIGRP neighbor, and router-id.
- **Interface**: Stores the configurations relevant to Interface such as summary-address, split-horizon, and authentication.
- **Topology**: Stores the configurations relevant to the EIGRP topology database such as administrative distance, redistribution, variance, and so on.

EIGRP Name mode configuration [1]*:*

```
router eigrp process-name
address-family {IPv4 | IPv6} {unicast | vrf vrf-name} autonomous-
system as-number
network network mask
```

5.4 Passive Interface

Passive interface configuration guarantees a level of security by stopping Send/Receive EIGRP Hello updates with a rogue router but still, the network on the interface will be advertised. The following is the passive-interface configuration on both EIGRP modes [35], [1]:

EIGRP Classic mode Passive-Interface configuration with "default" feature:

```
(config)#router eigrp as-number
(config-router)#passive-interface default (making all interfaces a
passive interface)
(config-router)#no passive-interface interface-type interface-val
```

EIGRP Name mode Passive-Interface configuration:

```
(config)#router eigrp process-name
(config-router)#address-family ipv4|ipv6 unicast autonomous-system as-
number
(config-router-af)#af-interface interface-type interface-val
(config-router-af-interface)#passive-interface
```

5.5 Authentication

Authentication is a security mechanism that ensures, EIGRP routers never receive malicious routing updates [1], [31]. EIGRP router sends the packets with a pre-computed password hash. The receiver decrypts the hash and compares the password if doesn't match then discards the packets [1].

EIGRP packet authentication is configured in two steps: A) The Configuration of a Keychain and key and B) The implementation of EIGRP authentication to use that keychain and key [31], [1]:

The first step of configuring EIGRP packet authentication:

- Specify a Key-Chain:

```
#key chain key-chain-name
```

- Configure a Key-ID under the Key-Chain. (It has to match on both side):

```
#key key-number
```

- Specifying a Key-String (Password) for the Key-ID (It has to match on both side):

```
#key-string password
```

The second step of configuring EIGRP packet authentication:

- Specify the hash-name: Message-Digest 5 (MD5) or Hashed Message Authentication Code-Secure hash Algorithm-256 (HMAC-sha-256) can be used on each interface of the connected peer:

```
#ip authentication mode eigrp as-number md5
```

- Association of the Key-string on each interface of the connected peer:

```
#ip authentication key-chain eigrp as-number key-chain-name
```

5.6 Load balancing

EIGRP supports load balancing over equal-cost (up to 4 equal-cost paths by default) and unequal-cost paths. In equal-cost path load balancing, EIGRP allows multiple successor routes to be installed into the routing table for the same prefix. The number of paths can be modified by the `#maximum-paths` command line under the EIGRP process in the Classic mode, and under the topology base submode in the Name mode [1], [33].
In Unequal-cost path load balancing, EIGRP allows both successor and feasible successor routes to be installed into the EIGRP routing table. This feature could be done by adjusting the Variance parameter's value (multiplier), which is set to 1 by default. If the

feasible distance of the feasible successor is less than the variance value then EIGRP allows it to be installed in the routing table [1], [34].

5.7 Modify Timers

EIGRP uses two time-intervals as per their specific requirements:

Hello-interval: To determine, how often hello packets are sent out of the interface. The second function of this time interval is to ensure the neighbors are still reachable [1].
Dead-interval (Hold-time): To determine, how long the router wait to react when no hello packets are received [36], [1].
These two time-interval don't need to match between neighbor routers. Their value is based on the speed of the interface [1]:

- T1 link or slower: Hello-interval is 60 sec and Dead-interval is 180 sec by default.
- LAN: Hello-interval is 5 sec and Dead-interval is 15 sec by default.

```
ip hello-interval eigrp as-number seconds
ip hold-time eigrp as-number seconds
```

5.8 Route summarization

Route summarization is used to reduce the routing table, reduces memory consumption, and CPU utilization. The router advertises the summarized network with the lowest metric belongs to the network within the summary range [2]. EIGRP installs a discard route that points to Null0, this is a routing loop-prevention mechanism [1]. In addition, route summarization is one of the methods to control query propagation in eigrp [1], [5]. EIGRP for IPv4 supports two types of route summarization , Interface -Specific summarization, and Automatic summarization [1].
Interface-Specific route summarization can be implemented in any part of the network and the summarizing router doesn't advertise the summarized network until a prefix matches it [1]. The following is Interface-Specific route summarization configuration command for both EIGRP Classic and Name modes:

Interface mode configuration of route summarization in EIGRP Classic mode:

```
ip summary-address eigrp as-number network subnet-mask
```

Interface mode configuration of route summarization in EIGRP Name mode:

```
summary-address network subnet-mask
```

Automatic summarization summarizes the classful networks by default but has been disabled since the release of IOS XE. This can be enabled with the auto-summary

command under the EIGRP process in classic mode, and under the topology base in the EIGRP name mode [1].

5.9 EIGRP Stub routers

To limit the EIGRP query scope we can configure a router as the stub router [5]. The EIGRP Stub router never advertises learned routes from one neighbor to another EIGRP neighbor, and non-stub routers never send the query to a stub router [5]. By default, a stub router only advertises connected, summary, and advertised routes [1]. The following command is used to configure the router as a Stub, this can be implemented under the EIGRP process for classic configuration and under the address family for named mode configuration [1]:

```
eigrp stub {connected | receive-only | redistributed | static | summary}
```

5.10 Route Filtering

EIGRP supports inbound and outbound traffic filtering on the interface. Route filtering is placed under the EIGRP process for the Classic mode, under the topology base for the Name mode. Filtering can be done with the following command [1]:

```
distribute-list {acl-number | acl-name | prefix prefix-list-name | route-map route-map-name} {in | out}
```

5.11 EIGRPv6

EIGRP for IPv6 has the same features and functions as EIGRP for IPv4 with the following general differences [32], [5]:

- First of all, the router must be enabled for ipv6 routing before configuring the routing process:

  ```
  (config)#ipv6 unicast-routing
  ```

- A 32-bit router-ID must be configured in case of not having an IPv4 address configured.
- EIGRPv6 routing process is a per-interface enabled configuration, EIGRPv6 has no network statement.
- EIGRPv6 updates will be sent using the interface's link-local address.
- An IPv6 keyword precedes in most of the commands instead of an IP keyword.
- IPv6 prefixes are encapsulated in the IPv6 packets.
- EIGRPv6 doesn't support the auto-summary feature.
- EIGRPv6 uses FF02::A as the multicast address.

The following is the EIGRP configuration for IPv6 in both Classic and Name modes [1]:

EIGRPv6 Classic Mode Configuration

Configure the EIGRPv6 process with the following global configuration command:

```
ipv6 router eigrp as-number
eigrp router-id
```

Enable the EIGRPv6 process on the interface:

```
ipv6 eigrp as-number
```

EIGRPv6 Name Mode Configuration

```
router eigrp process-name
address-family ipv6 autonomous-system as-number
eigrp router-id router-id
```

5.12 Troubleshooting

This section covers EIGRP network issues and troubleshooting tips. In addition, trouble ticket #4.1.2.3 from "Troubleshoot labs" has been worked at the end of this section, on behalf of other trouble tickets:

- It sometimes happens that the neighbor adjacency between two enabled EIGRP/EIGRPv6 routers is not formed, this may be due to [1]: **1)** Down status of the interface, **2)** Mismatched AS numbers (verified by #show ip protocols, #show ip eigrp neighbors), **3)** Bad network statement, **4)** Mismatched K-Values (verified by #show ip protocols), **5)** Passive interfaces, **6)** Configuring EIGRP routers on Different subnets, **7)** Timers, **8)** Mismatched Key string and key ID in Authentication, and **9)** ACLs.
- There are also problems such as not having EIGRP/EIGRPv6 routes in the routing table or topology table, it may due to [1]: **1)** Bad network statement, **2)** A more reliable source of information (better AD), **3)** Route Filtering (verified by #show ip protocol), **4)** Wrong Stub router configuration (verified by #show ip protocols, #show ip eigrp neighbors detail), **5)** Down status of the interface, and 6) Split Horizon, The Split Horizon feature is enabled by default for EIGRP and it may cause problems in networks like as DMVPN.

5.12.1 Ticket 4.1.2.3 (Troubleshoot EIGRP for IPv4)

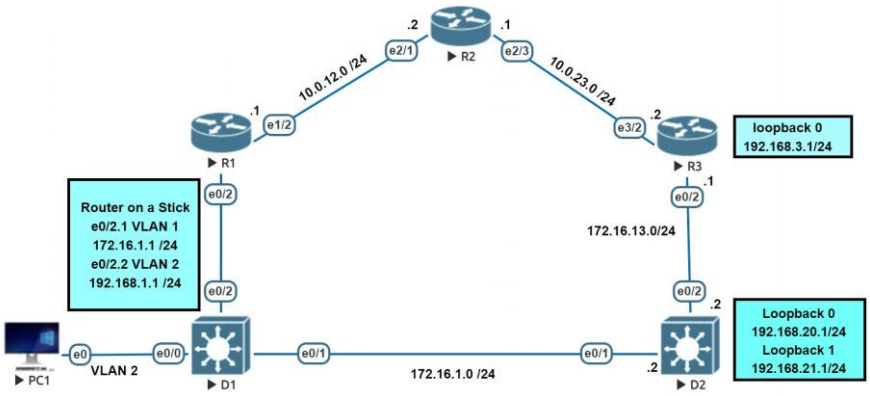


Figure 5. 4: Network topology for the emulated lab #4.1.2.3 "Troubleshoot EIGRP for IPv4"

In this ticket, Switch D2 was converted to support Inter-VLAN routing and configured to join the EIGRP domain but it is not forming adjacencies.
This problem could be due to incorrectly configured parameters or not configured at all (We assume that there is no Layer 1 and 2 issues in this network). We can start our troubleshooting step by step as per Cisco recommendation in section 5.12:

Switch D2 is not forming adjacencies:

```
D2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(421)
```

The show ip protocol command is used to debugging routing operations, which confirms that the routing protocol configuration works as expected. It provides information such as network statement, Router-ID, AS Number, K-values, Stub configuration, Route summarization, and filtering, see Figure 5. 5.

```
D2#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 421"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(421)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    Soft SIA disabled
    NSF-aware route hold timer is 240
  EIGRP NSF disabled
     NSF signal timer is 20s
     NSF converge timer is 120s
    Router-ID: 132.132.132.132
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    172.16.1.0/24
    172.16.13.0/24
    192.168.20.0
    192.168.21.0
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170
```

Figure 5. 5: Debugging routing operations for with show ip protocols command

We can now move on to the D2's neighbors. The table of neighbors is checked for R3 and R2, it is determined that they are configured with a different Autonomous System Number (ASN) than Switch D2:

```
R3#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(412)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   10.0.23.1               Et3/2                    11 00:54:13    9   100  0  8
```

```
R2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(412)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
1   10.0.23.2               Et2/3                    10 00:53:10    5   100  0  3
0   10.0.12.1               Et2/1                    12 00:53:55 1023  5000  0  5
```

D2 is configured with the wrong ASN 421, which is why a mismatched ASN is a barrier to EIGRP's adjacency. EIGRP AS #421 on switch D2 must be removed and reconfigured with all necessary configurations for AS #412:

```
D2(config)#no router eigrp 421
D2(config)#router eigrp 412
D2(config-router)#eigrp router-id 132.132.132.132
D2(config-router)#network 172.16.1.0 0.0.0.255
D2(config-router)#network 172.16.13.0 0.0.0.255
D2(config-router)#network 192.168.20.0 0.0.0.255
D2(config-router)#network 192.168.21.0 0.0.0.255
D2(config-router)#exit
D2(config)#interface e0/1
D2(config-if)#ip authentication mode eigrp 412 md5
D2(config-if)#ip authentication key-chain eigrp 412 security
D2(config-if)#exit
D2(config)#interface e0/2
D2(config-if)#ip authentication mode eigrp 412 md5
D2(config-if)#ip authentication key-chain eigrp 412 security
D2(config-if)#end
```

Verification of neighbor adjacency of D2 with D1 and R3:

```
D2#sh ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(412)
H   Address                 Interface              Hold Uptime   SRTT
RTO  Q  Seq
                                                   (sec)         (ms)
Cnt Num
1   172.16.13.1             Et0/2                    14 00:00:19    9
100  0  10
0   172.16.1.1              Et0/1                    10 00:00:20    9
100  0  9
```

6. OSPF

This chapter covers in more detail the laboratory material related to the OSPF routing protocol. The following labs are dedicated to the topics of this chapter:

- Lab **#6.1.2**_Implement Single-Area OSPFv2.
- Lab **#7.1.2**_Implement Multi-Area OSPFv2.
- Lab **#7.1.3**_OSPFv2 Route Summarization and Filtering.
- Lab **#8.1.2**_Troubleshoot OSPFv2 (It contains 3 trouble tickets 8.1.2.1-3).
- Lab **#9.1.2**_Implement Multiarea OSPFv3.
- Lab **#10.1.2**_Troubleshoot OSPFv3 (It contains 3 trouble tickets 10.1.2.1-3).

All the laboratories are emulated and available in the EVE-NG server, devices are configured with the initial configurations. Password on all devices is cisco12345, if a username is required, use the admin.

OSPF_ Open Shortest Path First (OSPF) is a Link State protocol that creates a Link-State Database (LSDB) of the network topology by sending the Link-state Advertisement (LSA) that includes link state and link metric. OSPF uses an algorithm called Dijkstra Shortest Path First (SPF) to find the best loop-free path. OSPF has an administrative distance of 110. This protocol provides more information about the network structure than other routing protocols, which makes better routing decisions [1], [12], [6].
To minimize the number of routing table records and the effect of network changes, OSPF divides the topology segment into areas within the routing domain. Routers or router interfaces located within the same area have the same Link-State DataBase (LSDB), this is because the neighbors are advertising the same link-sate information that the main router advertised it [2]. Each interface of a router can only participate in one area. As a result of this division, routers consume less memory and resources [1].
An OSPF network consists of a backbone area (area 0) where other non-backbone areas must be connected to the backbone area. The backbone area is designed as a transit zone, meaning that communication between other areas is possible through the backbone area [5]. The SPF algorithm runs in each area, while routers in one area have no information about LSDBs in other areas but only know about the networks [1].

By dividing an OSPF network into multi areas, new roles are defined for OSPF routers [5]:

- **Internal Router**: A router that has all its interfaces in a single area.
- **Backbone Router**: The Backbone router has at least one interface connected to Area 0
- **Area Border Router (ABR)**: A Router that has two or more connections to different areas.

- **Autonomous System Boundary Router (ASBR)**: ASBR is connected to OSPF and non-OSPF networks.

The OSPF network topology is divided into the following areas [1]:

- **Backbone Area**: It is also called Area 0; the other areas must be connected to it.
- **Regular Area**: non-backbone area whose database includes both internal and external routes.
- **Stub Area**: The database of this area only includes internal routes and a default route. Two types of LSAs (4 and 5) are not allowed in this area, so Area Border Router (ABR) generates a default route and advertises it [5].
- **Totally Stubby Area**: Cisco proprietary, its database contains its own area routes (intra-area) and a default route (Types 3,4, and 5 LSAs are not allowed).
- **Not-So-Stubby Area**: Its database includes internal routes (inter-are), redistributed, and Default routes.
- **Totally NSSA**: Cisco proprietary, its database contains its own area routes (intra-area), redistributed routes, and default routes.

For communication between routers, OSPF uses its own protocol number 89 and the following two multicast addresses as a destination [1]:

- **AllSPFRouters**: All OSPF-enabled routers must receive packets of this Multicast IPv4 address 224.0.0.5 (related MAC address 01:00:5E:00:00:05).
- **AllDRouters**: To communicate only with the Designated Router (DR) and Backup Designated Router (BDR) uses the IPv4 address 224.0.0.6 and related MAC address 01:00:5E:00:00:06.

OSPF uses five types of messages (packet) to communicate and maintain neighborship. OSPF sends the packets over the IP protocol using the OSPF Header [5]:

- **Hello**: Packets for identifying and maintaining the neighbors.
- **Database Description** (DBD): Packets for describing and summarizing database contents.
- **Link-State Request**: Packets for requesting the full information about the networks which the routers don't have, based on the received DBD packets.
- **Link-State Update**: Packets for the response of LSR packets.
- **Link-State Acknowledgment**: An LS acknowledgment packet will be sent to verify each newly received LSA.

The Hello packet is used to detect the neighbors, forming adjacency, and as a mechanism for announcing the existence and availability of any OSPF router. OSPF routers send these packets to the AllSPFRouters address (224.0.0.5). These packets are sent every 10 seconds, and if no packet is received within 40 seconds (Dead Time), the neighbor is

considered inactive and dead. The Hello packet contains the following data: Router ID, Authentication options, Area ID, Interface address mask, Interface priority, Hello and Dead interval, DR and DBR, Active neighbor. The following parameters must be considered to have an OSPF adjacency [1], [2]:

- Router IDs must be unique.
- Same subnet between two OSPF interfaces.
- Maximum Transmit Unit (MTU), Area ID, DR, Hello and Dead intervals, Authentication type, and Area type flags must match for the segment.

The following list describes the state of OSPF routers in the process of adjacency [11]:

- **Down**: The initial state that no Hello packet has been exchanged yet.
- **Init**: In this state, the Hello packet is received from another router, still no two-way communication
- **2-Way**: In this state, bi-directional communication is established. In addition, DR and BDR are also selected.
- **ExStart**: The starting point in the forming of an adjacency. In addition, Master and Slave are also selected for the LSDB synchronization.
- **ExChange**: In this state, the routers are started to exchange the link states by using the DBD packets.
- **Loading**: In this state, the router requests newer LSAs that were discovered but were not received in the Exchange state.
- **Full**: This is the last state in which neighbors have full adjacency and a synchronized LSDB.

The above OSPF neighbor state can be verified with the following command [1]:

```
show ip ospf neighbor
```

6.1 OSPF Configuration

This section covers the implementation of the OSPFv2 (OSPF for IPv4) and its features such as default router advertising, modifying link costs, placement of DR and BDR, etc. The following steps describe the implementation of the OSPF protocol on a router or multi-layer switch:

- Enable the OSPF process. A process id is a local number that does not need to be matched between neighbors. OSPF routers are allowed for multiple processes, but the routes in each OSPF process are advertised only by redistribution between processes [1]:

```
Router ospf <process-id>
```

- Enable OSPF on the interface (identifying the interfaces for the OSPF process) using the OSPF network statement (first method) under the OSPF process and specify the area in which the interface participates [1], [13] :

```
(config-router) # network ip-address wildcard-mask area area-id
```

Note = The primary IPv4 address and subnet mask of an interface to be selected for the OSPF process must match against network statements [1].

- Enable OSPF on the interface using the Interface-Specific method (second method) [1]:

```
(config)#interface interface-type interface-number
(config-if)# ip ospf process-id area area-id
```

Note1 = By default, OSPF advertises a 32-bit host route for interface loopback, to solve this, set the network type as a point-to-point [1], [14]:

```
(config)#int loopback 1
(config-if)#ip ospf network point-to-point
```

Note2 = The Interface-specific method takes precedence over the network statement method when both methods are configured at the same time [1].

6.1.1 Default Route Advertising

To advertise a default route into a normal area (Regular or Backbone Areas), the OSPF router must have a default route in the routing table that can be enabled by configuring a static default route. The advertiser router is known as the last resort gateway. The receiver router learns the route as an external OSPF Type 2 route, which is indicated by "O*E2" in the routing table. The OSPF router advertises the default route with the following command [1], [15]:

```
default-information originate [always] [metric metric- value] [metric-type type-value]
```

[**always**]: OSPF can advertise a default route even if it does not exist in the routing table

6.1.2 Link Costs

To find the best route metric, OSPF calculates the total cost of all OSPF output interfaces in that route then picks the route with the lowest total cost as the best route metric [5]. This is because interface cost is inversely related to the interface bandwidth [16]:

```
Cost = Reference bandwidth / Interface bandwidth
```

By default, Reference bandwidth has a value of 100 Mbps which is unable to differentiate interfaces faster than 100 Mbps (FastEthernet) then OSPF assigns the identical cost of 1

[1]. This is why Cisco allows us to modify the link cost by changing the Reference bandwidth value with the following command:

```
auto-cost reference-bandwidth reference-bandwidth
```

Note = It is recommended to configure the above command with the same value on all OSPF routers within the area [1].

6.1.3 DR and BDR Election

When the number of routers increases in Multi-access networks such as LAN, in parallel the number of neighbor adjacencies also increases. As a result, with more adjacency and more LSA flooding, the router consumes more bandwidth, memory, and CPU that is leading to scalability problems. To avoid this, communication and exchange of information should not be done in each pair. A router with a higher priority or higher Router ID (RID) is selected as the Designated Router (DR). The DR task is to keep all routers up to date. There is a problem with this design, in case of a failed DR the network will be disrupted. To prevent this problem, another router is considered as a Backup Designated Router (BDR) and constantly checks the existence of the DR and in case of any problem with DR, it will be selected as the DR. Other routers called DROTHER, have only a full adjacency with DR and BDR [1], [2], [16].

6.1.4 Exploring Link State Announcements

The router connected to each area creates a unique LSDB for that area, which contains the latest received LSA. The ABR keeps the LSAs separate for each area in a different LSDB. Each LSA has a sequence number and age of 30 minutes by default. If the LSA age reached 30 minutes, its origin router will flood a new LSA with the age value of 0 which increments by 1 every second. When an LSA is received, it is compared to the LSDB database. If the LSA is new, it is added to the database then the SPF algorithm is executed. If the LSA is from a router that already exists in the database, its Sequence Number is compared, in case of a lower number, it will be ignored [1]. The OSPF protocol includes different types of LSAs, in the following each of them will be described in detail:

- **LSA type 1** (Router Link): Each OSPF enabled interface generates and advertises a Type 1 LSA in that area that is not allowed to cross the area [1], [5].
  - It can be verified by **show ip ospf database router** command [1].
- **LSA type 2** (Network Link): This type of LSA is always generated and advertised by DR for all OSPF-enabled routers in that area. it describes the network segment, the neighbors connected to that segment, and the network subnet mask [1], [5].
  - It can be verified by **show ip ospf database network** command [1].

- **LSA type 3** (Summary Link): LSA types 1 and 2 are not allowed to be advertised out of the origin area, so this is the ABR responsibility that creates a type 3 LSA and locates the networks from LSA types 1 and 2 inside the type 3 LSA and advertises it to another area [1], [5].
  - It can be verified by show ip ospf database summary command [1].
- **LSA type 4** (ASBR Summary): This type of LSA is generated by ABR when it received the special LSA type 1 from ASBR (External Routes). LSA type 4 advertises the existence of ASBR for other areas [1], [5].
  - It can be verified by show ip ospf database asbr-summary command [1].
- **LSA type 5** (External Routes): To advertise the OSPF external routes to other areas, ASBR generates the LSA type 5 and advertises it throughout the entire OSPF domain [1], [5].
  - It can be verified by show ip ospf database external command [1].
- **LSA type 7** (NSSA External Summary): This type of LSA is generated by ASBR when it injects the OSPF external routes to a Not-So-Stubby Area (NSSA). LSA type 7 is not allowed to leave the NSSA, so the ABR changed the LSA type 7 to LSA type 5. The LSA type 5 also will be changed to the LSA type 4 by the next ABR after the backbone area. The only difference of Type 7 LSA field in comparison to Type 5 LSA is in Metric type, Type 7 LSA includes two types of external metric: A) Type 1 (O N1) and B) Type 2 (O N2) [1], [5].
  - It can be verified by **show ip ospf database nssa-external** command [1].

6.1.5 Route Summarization

Because each router within an area must maintain the same copy of LSDB, only ABR and ASBR are allowed to perform Route Summarization, which makes the Route Summarization for interarea routes on ABR and external routes on ASBR. Interarea summarization reduces the number of LSAs type 3 and can be implemented using the following command [1], [2]:
Note = ABR installs discard routes to null for preventing the routing loops.

```
area area-id range network subnet-mask [advertise | not-advertise]
[cost metric]
```

The following command implements route summarization on an ASBR router:

```
summary-address network subnet-mask
```

Less memory consumption, effective CPU utilization, and faster SPF calculation can be counted as the most advantages of route summarization within OSPF enabled network. In addition, route summarization cancels the SPF calculation of summarized routes in non-origin areas which further helps to save resources when a link is flapping [5].

6.1.6 Route Filtering

OSPF supports the three most common techniques for route filtering which are generally implemented on ABR [1], [2]:

- **Route Filtering with summarization no-advertise**: ABR stops LSA type 3 creation by doing this filtering method. This method can be implemented using the route summarization command by adding the **not-advertise** keyword at the end of the command which filters any networks in LSA type 3:

```
area area-id range network subnet-mask not-advertise
```

- **OSPFv2 area filtering**: This filtering method can be occurred using the Filter-list tool. Area filtering is implemented on ABRs, it can filter various routes located in the LSA type 3. Area filtering can occur on both input and output interfaces:

```
area area-id filter-list prefix prefix-list-name [in|out]
```

- **Local OSPFv2 filtering**: The local filtering prevents the topology table routes from being installed in the local routing table, which can be implemented with the following command:

```
distribute-list {Acl-number | ACL-name | prefix prefix-list-name
| route-map route-map-name} in
```

6.2 OSPFv3

OSPFv3 has been developed with some similarities and differences compared to OSPFv2. OSPFv3 still has the same concept of area, metric (cost), SPF, election process of DR and BDR, same packet (message) types, neighbor discovery mechanism, OSPF network types, etc [8]. OSPFv3 differs from OSPFv2 in the following features: **a)** support multiple Address Family (AF) IPv4 & IPv6, **b)** runs over IPv6, **c)** link-local IPv6 address is used as the source for inter-router communication, **d)** terminology of 'link' instead of 'network', **e)** manual specification of router-id and neighbor, **f)** authentication using IPsec extension headers, **g)** 2 new LSAs to advertise IP address information, **h)** support of multiple instances per link, **i)** support of neighbor adjacency in different subnets [1].
In order to have inter-router communication, OSPFv3 uses its own protocol number 89 (same as the OSPFv2) and as the destination address uses either a unicast address or the following two multicast addresses: a) FF02::05: OSPFv3 AllSPFRouters, b) FF02::06: OSPFv3 AllDRouters designated router (DR) router [2].

6.2.1 OSPFv3 LSA types

OSPFv3 designers have made this version of OSPF more efficient by introducing two new types of LSAs. Prefix information is no longer advertised by the LSA type 1, but only the type and cost of the interface, this separates the SPF tree and the prefixes, so

there is no need to calculate the full SPF tree each time. The IP address information is advertised using new types of LSAs **a) Type 8 LSA** (Link-Local LSA): Link-local addresses are advertised in Type 8 LSA between neighbors on the same link. This type of LSA also maps the global unicast address prefix to the link-local address associated with an interface, **b) Type 9 LSA** (Intra-area prefix LSA): Prefixes are advertised in Type 9 LSA. The following are the types of all LSAs in OSPFv3: Type **1)** Router, Type **2)** Network, Type **3)** Inter-area Prefix, Type **4)** Inter-area router, Type **5)** AS-external, Type **7)** NSSA, Type **8)** Link-local, and Type **9)** Intra-area prefix. The ASBR name in OSPFv3 is also changed to the inter-area router LSA [1], [18].

6.2.2 OSPFv3 Configuration

To implement the OSPFv3, first of all, it needs to enable IPv6 routing using the following *ipv6 unicast-routing* command. After that, OSPFv3 is configured using the command ***ospv3*** [*process-id*]. OSPFv3 configuration also needs to specify a unique 32-bit value router-id. The next step is the initialization of AF within the routing process using command ***address-family*** *{**ipv6** | **ipv4**} unicast.* The last step is enabling the protocol for an interface and assigning the interface to an area within interface command ***ospfv3*** *process-id* ***ipv6 area*** *area-id.* OSPFv3 has the same configuration as OSPFv2 for the features such as the passive-interface configuration, route summarization, default route advertising, etc. Regarding the verification commands (show) is only necessary to change the ip for ipv6 e.g., ***show ipv6 ospf neighbor*** [1],[2], [17].

6.3 Troubleshooting

This section covers problems you may encounter when implementing OSPF and the troubleshooting techniques to solve these problems. In addition, trouble ticket #10.1.2.2 from "Troubleshoot labs" has been worked at the end of this section, on behalf of other trouble tickets.

- Regarding troubleshooting OSPF neighbor adjacency, it should be considered more about the layer 1 connectivity, misconfiguration of interface for the OSPF process, mismatched area numbers, mismatched timers, neighbor adjacency with different subnets (only for OSPFv2), MTU mismatched, mismatched authentication information, and No ACL is blocking packets towards the multicast address of 224.0.0.5 [1].

- Regarding troubleshooting OSPF routes that may be missing, it should be considered more about the route filtering, misconfiguration of interface for the OSPF process, stub area configuration, unique RID configuration for each OSPF enabled router, the status of layers 1 and 2 (up/up) for the participated interfaces in the OSPF process, and more reliable source of information [1], [19].

- The following are the most useful verification commands for OSPF troubleshooting [1]:

```
show ip ospf interface [brief|interface-id] verify the OSPF-enabled interface

show ip ospf neighbor [detail] verify the OSPF neighbor adjacency

show ip route ospf verify OSPF installed route in the routing table

show ip ospf topology  verify the OSPF topology table
```

6.3.1 Trouble ticket #10.1.2.2 (OSPFv3)

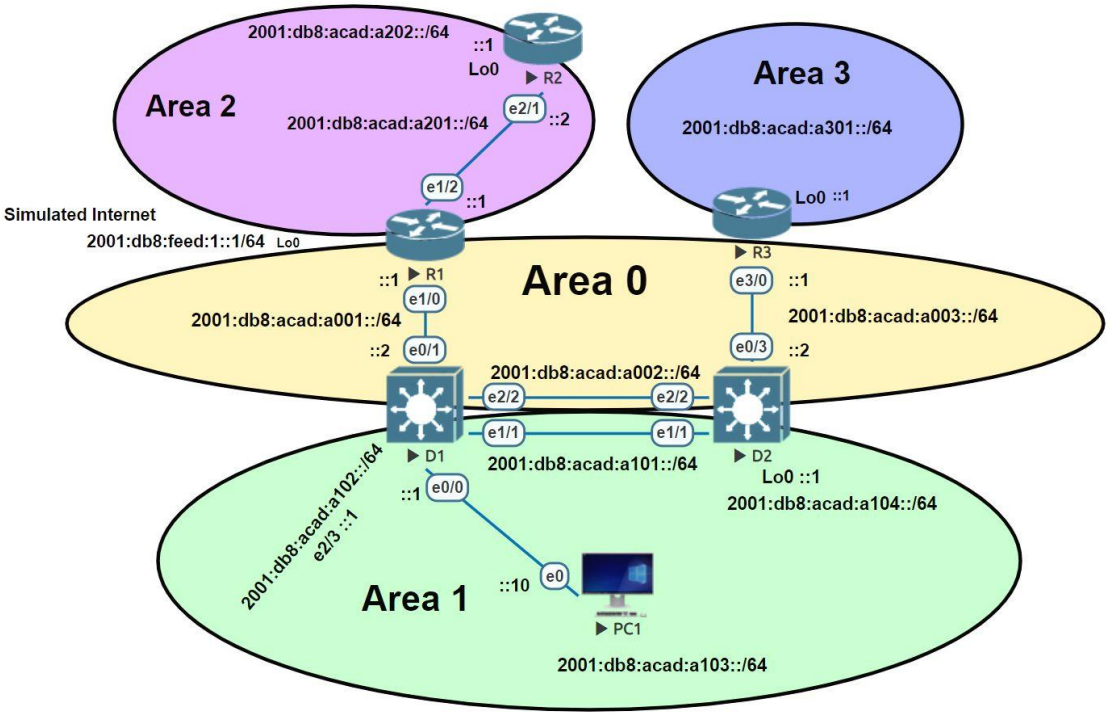


Figure 6. 1: Network Topology for the emulated lab "Troubleshoot OSPFv3"

After network maintenance, users started reporting that there is no access to the IPv6 prefix in area 3. You have been selected to find and resolve the issue, all devices should be able to ping 2001:db8:acad:a301::1/64.

First of all, the topology in Figure 6. 1 shows that there is an OSPFv3 network where all devices are configured with an IPv6 address.

According to cisco's recommendation, the Follow-the-Path troubleshooting method is used to test PC1 access to network resources in Area 3 [10], see Figure 6. 2. It confirmed that there is no OSPFv3 connection between D2 and R3 from PC1. The next step is to jump on D2 to check the OSPFv3 Neighbor table, see Figure 6. 3.

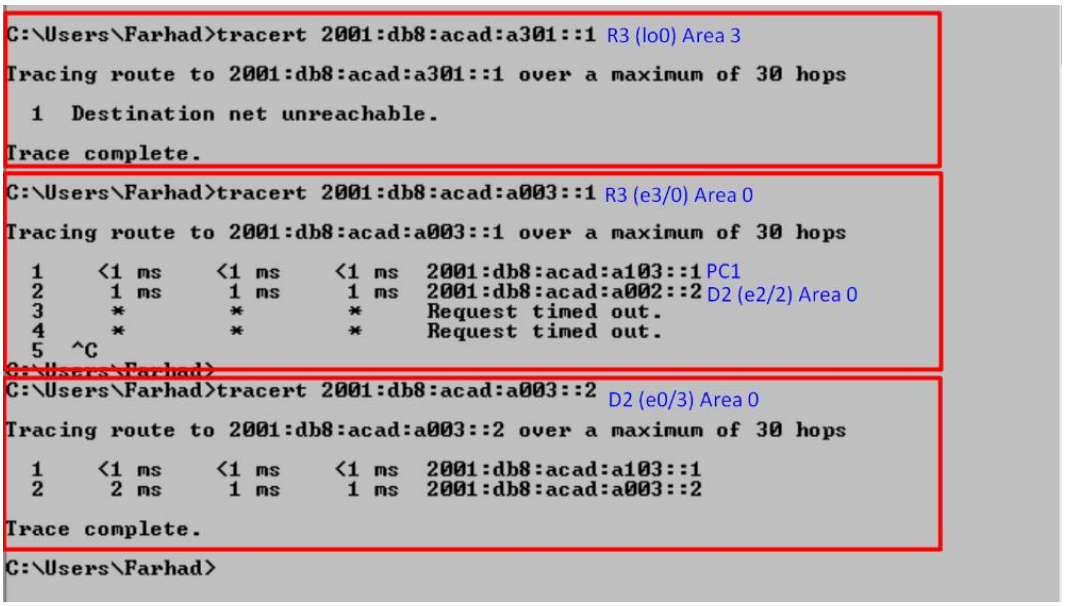


Figure 6. 2: Follow-the-Path troubleshooting approach, testing the reachability from PC1 to the link between R3 and D2 (Area 0)

```
D2#sh ipv6 ospf neighbor

            OSPFv3 Router with ID (5.5.5.5) (Process ID 1)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
4.4.4.4 D1        1   FULL/BDR        00:00:36    12              Ethernet2/2
4.4.4.4           1   FULL/BDR        00:00:36    7               Ethernet1/1
```

Figure 6. 3: OSPFv3 neighboring table on D2

Switch D2 has only neighbor adjacency with D1 in two difference Areas (0 and 1), It can also be verified that no route from area 3 is learned, see the D2 routing table in Figure 6. 4:

```
   Inter-area Route List

*>  2001:DB8:ACAD:A201::/64, Inter, cost 30, area 0
      via FE80::A002:1, Ethernet2/2
*>  2001:DB8:ACAD:A202::/64, Inter, cost 31, area 0
      via FE80::A002:1, Ethernet2/2
```

Figure 6. 4: D2 routing table

Now it's time to get into R3 to verify that the routing protocol configuration works as expected. The show ip protocol command is used to debugging routing operations, it provides information such as network statement, Router-ID, OSPF process number, area number, Stub configuration, Route summarization, and filtering, see Figure 6. 5:

```
R3#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "application"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 1"
  Router ID 0.0.0.0
  Number of areas: 2 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    Ethernet3/0
  Interfaces (Area 3):
    Loopback0
  Redistribution:
    None
```

Figure 6. 5: Debugging routing operations with show ip protocols command.

It is verified that R3 has no configured router-id. R3 does not have any interfaces with IPv4 addresses, the OSPF process was unable to create a 32-bit router ID, this can also be confirmed by checking the R3 neighbor table, see Figure 6. 6

```
R3#show ipv6 ospf neighbor
%OSPFv3 1 address-family ipv6 not running, please configure a router-id
```

Figure 6. 6: Verify the OSPFv3 neighbor table on R3 by using the show ipv6 ospf neighbor command.

Now it is time to configure a unique router-id on R3 for OSPFv3 process 1 [19]:

```
R3(config)#router ospfv3 1
R3(config-router)#address-family ipv6 unicast
R3(config-router-af)#router-id 3.3.3.3
R3(config-router-af)#
*May 13 05:32:20.217: %OSPFv3-5-ADJCHG: Process 1, IPv6, Nbr 5.5.5.5
on Ethernet3/0 from LOADING to FULL, Loading Done
```

PC1 now has access to network resources in Area 3:

```
C:\Users\Farhad>ping 2001:db8:acad:a003::1

Pinging 2001:db8:acad:a003::1 with 32 bytes of data:
Reply from 2001:db8:acad:a003::1: time=1ms
Reply from 2001:db8:acad:a003::1: time=1ms
Reply from 2001:db8:acad:a003::1: time=1ms
Reply from 2001:db8:acad:a003::1: time=2ms

Ping statistics for 2001:db8:acad:a003::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

7. BGP

This chapter covers the laboratory task topics related to the BGP routing protocol. The following labs are dedicated to the topics of this chapter:

- Lab **#11.1.2**_Implement eBGP for IPv4.
- Lab **#11.1.3**_Implement MP-BGP.
- Lab **#12.1.2**_Implement BGP Path manipulation.
- Lab **#13.1.2**_Implement BGP Communities.
- Lab **#14.1.2**_Troubleshoot BGP (It contains 2 trouble tickets 14.1.2.1-2).

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

Border Gateway Protocol (BGP) is a path-vector routing protocol that falls into the category of External Gateway Protocols (EGPs). This is a protocol used for Internet routing, routing between service providers, or rather routing between different numbers of the autonomous system [5].
BGP consists of two types of sessions: **A)** BGP session to a neighbor in the same autonomous system is called internal BGP (iBGP) which has an Administrative Distance (AD) of 200. **B)** A BGP session with a neighbor in a different AS (between autonomous systems) is called an external BGP (eBGP) which has an AD of 20 [1].
BGP peers perform a three-way TCP handshake and use TCP port number 179 before establishing adjacency [51].
BGP has the capability of having Single-hop and Multi-hop sessions. The single-hop session is about the adjacency between directly connected routers, often used by IGPs. BGP can also have a single-hop session but without exchanging Hello packets compared to IGP. The eBGP is mostly used for a single-hop session because its Time-To-Live (TTL) is set to 1. To have a multi-hop session, TTL should be more than 1, and BGP needs routing table information (provided by the IGP or Static route) to establish a TCP session. iBGP is more suitable for a multi-hope session because its TTL is set to 255 by default [1], [2].
External BGP neighbors use the AS-Path attribute (well-known mandatory) as a loop-prevention mechanism and shortest path determinant. AS-Path attribute provides a sequential list of all AS numbers through which the advertised prefix is passed. The path with the lowest AS number is selected as the shortest path. If the BGP neighbor finds its own AS number in this list, it will discard the advertised prefix to prevent the routing loop. While there is no way to identify the routing loop for iBGP neighbors, they are not allowed to advertise the received prefix to each other, so all iBGPs must fully mesh in the same AS [1], [51].

BGP for IPv6 can be implemented through Multi-Protocol BGP (MP-BGP) (RFC 2858). The RFC 2858 has been released to introduce MP-BGP by adding two extensions, **1)** Address Family Identifier (AFI), **2)** Subsequent Address Family Identifier (SAFI). AFI is developed to identify specific network protocols, such as IPv4 or IPv6 on the other hand, SAFI identifies unicast, multicast, MPLS Label, and MPLS Label VPN. BGP has two BGP path attributes within the update message that are used to transmit Network Layer Reachability Information (NLRI) for each Address Family (AF). There is a separate database and configuration for each Address Family [1], [5].

BGP Messages:

BGP routers use the following 4 types of messages to communicate with each other [1], [51]:

- **Open message**: The open message includes hold time (180 seconds by default), BGP (router ID), BGP version, and ASN. To have a BGP adjacency, both sides exchange the Open messages the BGP versions and ASN must much. In addition, each router must have a unique BGP identifier (router-id).
- **Keepalive message**: BGP routers send these types of messages to check the availability of neighbors. The default time interval for the Keepalive message is 60 seconds (one-third of the maintenance time).
- **Update message**: This type of message is exchanged between BGP routers to obtain information about backup paths, failed routes, and Network Layer Reachability Information (NLRI) that includes its BGP Path Attribute and prefix. Receiving any Update or Keepalive message resets the Hold time.
- **Notification message**: BGP routers use this type of message to notify each other of any errors that are detected.

BGP maintains the table of all BGP neighbors and their operational status using the Finite-State Machine FSM [2]. FSM can be in exactly one of the following states for BGP neighbors [1], [51]:

- **Idle**: This is the first state of BGP neighborship. In this state, the BGP router is waiting for the "Event Start", the Event Start occurs with a new BGP neighbor configuration or reset of the current BGP neighbor status. When the Event Start occurred, the BGP router starts the ConnectRetry timer, tries to initiate a TCP connection, and listens to any TCP connection from other peers. If the neighboring state returns to Idle with any network error, the next TCP connection attempt will take approximately after 60 seconds.
- **Connect**: In this state, if a three-way TCP handshake is performed, the BGP router clears the timer, sends an open message, and moves into OpenSent state, but in the opposite case, with a failed TCP three-way handshake, it will move to Active state and will wait for a TCP connection from the neighbor in the Active State. If

the ConnectRetry timer expires during Connect state, it stays in Connect state and tries again for a successful TCP three-way handshake. The neighboring state returns to the Idle state after receiving any other errors.

- **Active**: In this state, if the TCP three-way handshake is performed then the BGP router resets the time, sends an Open message, and moves to the OpenSent state. If the ConnectRetry is expired during this state, the state returns to the Connect state. The neighboring state returns to the Idle state after receiving any other errors.
- **OpenSent**: In this state, BGP routers check open messages for errors such as incompatibility with BGP versions, incorrect ASN, incorrect configuration of the source IP address, and so on. If an error is found, the router sends a Notification message in response to the error and returns to Idle mode. If there is no error in the Open message, the BGP router clears the Keepalive timer and sends the Keepalive message, and goes into OpenConfirm state. In this state, before moving to the OpenConfirm state, the BGP routers negotiate for the hold timer (lower value) and determine the BGP sessions (iBGP or eBGP). If the TCP session fails in OpenSent state, the BGP neighboring state returns to Active state. The neighboring state returns to Idle mode after receiving any other errors.
- **Open Confirm**: In this state, after receiving the Keepalive message from the neighbor, the BGP state will move to the Established state, but in the opposite case, the neighboring state will return to the Idle state if it receives the Notification message. The neighboring state also returns to the Idle state after receiving any other error.
- **Established**: In this state, the BGP routers are completely adjacent and start exchanging update messages. Receiving Updates or Keepalive messages resets the hold timer. The neighboring state returns to the Idle state after receiving a notification message or any other error.

7.1 Implementation of BGP for IPv4

External BGP (eBGP) requires two Cisco IOS routers to be directly connected (TTL=1) to establish a neighbor adjacency otherwise the next-hop address reachability is needed [52]. The eBGP adjacency can be established between two routers with the following CLIs [1]:

Initiate the BGP process:

```
#router bgp <AS#>
 #bgp router-id <router-id>
```

Identify the BGP peer's IP address and its AS number:

```
 #neighbor <neighbor's IP address> remote-as < remote-as number>
```

To install the routes in the BGP table, the exact matching of the network and the subnet mask in the routing table are required.

```
 #network <prefix> mask <network mask>
```

IOS enables IPv4 AF by default. otherwise, the command must be executed to disable:

```
#no bgp default ipv4-unicast
```

Address Family can be activated with the following command under BGP process:

```
#address-family {ipv4|ipv6}
  #neighbor <ip-address> activate
```

In the case of iBGP adjacency, neighbors must be configured with the same ASN, and neighbors must be configured with the next-hop-self feature to ensure that the next-hop address check is passed without advertising prefixes into BGP [5]:

```
neighbor ip-address next-hop-self [all]
```

7.2 Implementation of MP-BGP

All BGP features and rules for IPv4 also apply to MP-BGP, the only difference being that initialization of IPv6 address family and neighbor activation is required. IPv4 or IPv6 addresses can be used to set up a TCP session. According to Cisco, unique global unicast addresses should be used for adjacency instead of link-local addresses [1]:

```
#router bgp AS#
 #bgp router-id router-id
 #neighbor < neighbor's IPv6 address > remote-as < remote-as number>
#address-family ipv6
 #neighbor < neighbor's IPv6 address > activate
```

7.3 Route Aggregation

Route aggregation (Route Summarization) is a method to reduce the size of the routing table that helps calculate the best path, less router resource consumption, and more stability [20]. Route aggregation can be implemented with two techniques of Static and Dynamic. (Dynamic route aggregation is discussed here). In both methods, the router installs the discard route for the summary route to prevent the routing loop [1].
BGP peers insert one of the following two Path Attributes to the aggregated route during route aggregation:

- **Aggregator Attribute**: (Optional-Transitive) Specifies the router ID and Autonomous System Number (ASN) of the aggregator router [51].

- **Atomic-Aggregate**: (Well-Known Discretionary) When AS-SET is not added to the path, this path attribute is inserted to verify that this route is a summary route and all specific subset information is lost [51].

Dynamic route aggregation is implemented with the following command [1]:

```
aggregate-address network subnet-mask [summary-only] [as-set]
```

- The keyword **summary-only** is used to suppress the original component network (Atomic-Aggregate path attribute will be attached to the aggregated route).
- To advertise the aggregated prefix without losing path information, the **as-set** keyword must be added to the aggregate-address command.

7.4 Default Route Advertising

The most significant way to advertise a default route in BGP is using the following common, it advertises the default route only to a specific neighbor [1]:

```
#router bgp <ASN>
 #neighbor <ip-address> default-originate
```

7.5 BGP Route Filtering

This section covers three common methods to filter routes for BGP: **1)** Distribute-List, **2)** Prefix-List based, **3)** AS-Path ACL.

7.5.1 Distribution List filtering

The Distribution List filtering uses the ACLs to filter the BGP routes on a neighbor-by-neighbor basis [1]. Standard ACLs can only match the prefix, while extended ACLs are used to match the prefix and subnet mask. Extended ACLs have a different mechanism for filtering the route in BGP, the source field is used to match the network and the destination field is used to match the subnet mask [51], see Figure 7. 1.

Figure 7. 1: The concept of extended ACL is used to filter the BGP path.

The following BGP address-family configuration command is used to implement the Distribution List filtering [1]:

```
neighbor ip-address distribute-list {acl-number | acl-name} {in | out}
```

7.5.2 Prefix-list based route filtering

The Prefix-List filtering uses the Prefix-List to filter the BGP routes on a neighbor-by-neighbor basis [1]. Prefix lists are used to match a network with a specific prefix length

or a range of prefix lengths. The following BGP address-family configuration command is used to implement the Prefix List filtering [1]:

```
neighbor ip-address prefix-list prefix-list-name {in | out}
```

7.5.3 AS-Path ACL to filter routes being advertised

To filter routes based on as-path, Regular Expressions (Regex) are required to match [51]. The following Table 7. 1 provides a list of common regular expressions [1]:

Modifier	Description
_	Matches a space
^	Indicates the start of the string
$	Indicates the end of the string
[]	Matches a single character or nesting within a range
-	Indicates a range of numbers in brackets
[^]	Excludes the characters listed in brackets
()	Used for nesting of search patterns
|	Provides or functionality ro the query
.	Matches a single character, including space
*	Matches zero or more characters or patterns
+	Matches one or more instances of the character or pattern
?	Matches one or no instances of the character or pattern

Table 7. 1: Regex Query Modifiers

The AS_Path filter uses the AS_Path access list to select routes. The following commands show, how to define the AS_Path ACL and how to implement it respectively [1]:

```
ip as-path access-list acl-number {deny | permit} regex-query
neighbor ip-address filter-list acl-number {in | out}
```

7.6 Path attribute manipulation

Each network path comes with explanatory characteristics called Path Attributes (PAs). Path features are commonly used to identify loop routing and determine the best path for a network prefix [5].

The following Table 7. 2 presents the classification of BGP prefixes [1]:

Name	Supported by All Vendors	Advertised Between AS
Well-known mandatory	Yes	Yes
Well-known discretionary	Yes	No
Optional transitive	No	Yes
Optional nontransitive	No	No

Table 7. 2: BGP Path Attribute Classifications

BGP Best Path Selection Algorithm uses the following list of attributes to select the best path for the same network prefix [1]:

Priority	Attribute
1	**Next-hop IP address** reachability (Mandatory PA)
2	**Weight** = The highest value is more preferable
3	**Local Preference** = The highest value is more preferable
4	**Originate** = Local route is more preferable
5	**AS path length** = Shortest length is more preferable
6	**Origin Code** = Lowest is more preferable IGP < EGP < INCOMPLETE (redistribute/aggregation)
7	**MED (Metric)** = Lowest is more preferable
8	**eBGP** path over **iBGP** path is more preferable
9	The **Shortest IGP path** to BGP next hop is more preferable
10	**Oldest Path**
11	**Router ID** = Lowest ID is more preferable
12	**Neighbor IP address** = Lowest address is more preferable

Table 7. 3: The list of BGP attributes in order, to select the best BGP Path.

7.7 BGP community

BGP community is an optional transitive Path Attribute used to tag the routes for easier control over routing policies and information in complex scenarios [1]. BGP community is found in a 32 bits value number or two 16 bits values AA: NN (new format) [1]. The following global configuration command is used to enable the use of the new BGP community format and deploying a BGP community on a route [51]:

```
1)ip bgp-community new-format
2) Use a route map to specify the routes.
3) Use the set community <community-valu> command to set the value for
the identified routes inside the route map.

4) #router bgp <ASN>
    #Neighbor <neighbor-address> send-community
Send-community statement sends the community attribute to the BGP
neighbor.
```

The following are the most common well-known communities, defined in RFC 1997 [1], [51]:

- **Internet**: This well-known BGP community is used to advertise the BGP route to all BGP peers.

- **No_Advertise**: This BGP community is used to do not advertise the BGP route. It can be implemented locally for inbound traffic or on the upstream peer for outbound traffic. The following command can be used to configure the No-Advertise community: **#set community no-advertise**.
- **No_Export**: This BGP community is used to do not advertise the BGP route to any eBGP peer. It can be implemented locally for inbound traffic or on the upstream peer for outbound traffic. The following command can be used to configure the No-Export community: #**set community no-export**.
- **Local_AS**: This well-known BGP community is used to do not advertise the BGP route out of the local AS. The Local-AS community can be implemented by the following command: **#set community local-as**

7.8 Troubleshooting

This section covers problems you may encounter when implementing BGP and the troubleshooting techniques. In addition, trouble ticket **#14.1.2.1** from "Troubleshoot labs" has been worked at the end of this section, on behalf of other trouble tickets.

- Regarding troubleshooting BGP neighbor adjacency, it should be considered more layer 1 and 3 connectivity, neighbor connectivity with no default route, correct configuration of the source IP address, mismatch authentication, and no ACL is blocking TCP port 179. The following command is used to show the bgp neighbor status [1]:

```
show bgp [ipv4|ipv6] unicast summary
show [ipv4|ipv6] bgp summary
show bgp [ipv4|ipv6] unicast neighbors
```

- Regarding troubleshooting BGP routes that may not be present in the routing or BGP table, it should be considered more about Next-hop router reachability, network mask command, route filtering, not sharing the iBGP-learned routes to another iBGP neighbor and more reliable source of information. The following command verifies the BGP-learned routes inside the BGP table [1]:

```
show bgp [ipv4|ipv6] unicast
show [ipv4|ipv6] bgp
```

- Regarding troubleshooting BGP path selection, it should be noted that BGP is a path-vector routing protocol, which means that BGP uses different path attributes instead of the interface's bandwidth to decide on the best route. CISCO IOS reviews BGP attributes to decide on the best route, highest weight, highest local preference, etc [1].

7.8.1 Ticket 14.1.2.1 (MP-BGP)

In this section trouble ticket **#14.1.2.1** from “Troubleshoot labs” has been worked, on behalf of other trouble tickets.

In this trouble ticket, recently R2 has been added between routers R1 and R3. After this new implementation, RIB indicates that both R1 and R3 have only BGP routes that are directly connected to their eBGP peers:

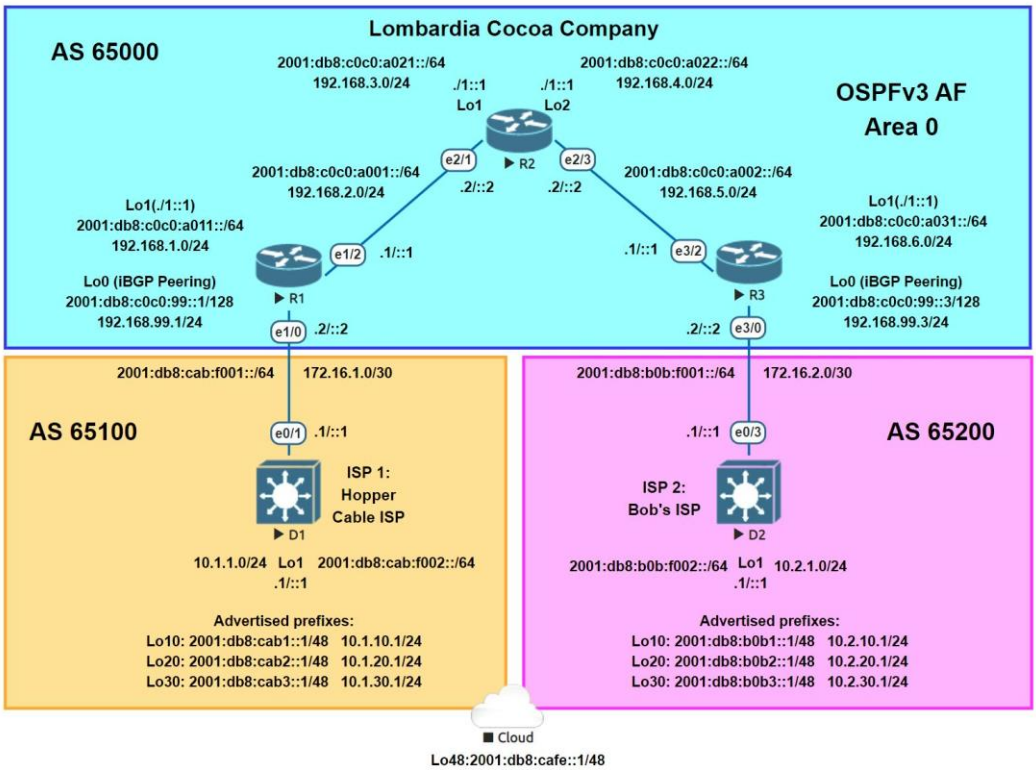


Figure 7. 2: Network Topology for the emulated lab "Troubleshoot BGP"

```
R1#sh ip route bgp
……
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 4 subnets
B        10.1.1.0 [20/0] via 172.16.1.1, 00:22:16
B        10.1.10.0 [20/0] via 172.16.1.1, 00:22:16
B        10.1.20.0 [20/0] via 172.16.1.1, 00:22:16
B        10.1.30.0 [20/0] via 172.16.1.1, 00:22:16
```

When we checked the BGP table on R1, found that networks in AS # 65200 were advertised but failed to choose the best routes (it is valid "*****" but not best ">"), because the next hop is set 172.16.2.1 and R1 doesn’t know where this IP is located:

```
R1#sh ip bgp
……
     Network          Next Hop            Metric LocPrf Weight Path
```

```
 * i 10.2.1.0/24      172.16.2.1               0    100      0 65200 i
 * i 10.2.10.0/24     172.16.2.1               0    100      0 65200 i
 * i 10.2.20.0/24     172.16.2.1               0    100      0 65200 i
 * i 10.2.30.0/24     172.16.2.1                0    100      0 65200 i
……..
```

According to Cisco recommendations, there are three ways to prevent this issue: **a)** IGP advertisement, **b)** Advertise the networks into BGP, **c)** Managing next-hop IP addresses with configuring next-hop-self feature. The third way is selected, R1 and R3 will be configured with the next-hop-self feature to pass the next-hop address check.

Note = The following is an example of the next-hop-self feature configuration on R1 only for IPv4 network, both R1 and R3 should be configured with this feature for IPv4 and IPv6 networks:

```
R1(config)#router bgp 65000
R1(config-router)#address-family ipv4 unicast
R1(config-router-af)#neighbor 192.168.99.3 next-hop-self
```

It is now, AS **# 65200** networks are advertised, they are valid (*****) and are selected as the best route (>) with 192.168.99.3 as the next-hop IP address:

```
R1#sh ip bgp

 *>i 10.2.1.0/24      192.168.99.3             0    100      0 65200 i
 *>i 10.2.10.0/24     192.168.99.3             0    100      0 65200 i
 *>i 10.2.20.0/24     192.168.99.3             0    100      0 65200 i
 *>i 10.2.30.0/24     192.168.99.3             0    100      0 65200 i

R1#ping 10.2.1.1 source lo1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.1.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

8. ROUTE MAPS AND CONDITIONAL FORWARDING

This chapter covers in more detail the laboratory material related to Route Maps and Conditional Forwarding. The following labs are dedicated to the topics of this chapter:

- Lab **#15.1.2**_Control Routing Updates.
- Lab **#15.1.3**_Path Control Using PBR.
- Lab **#15.1.4**_Troubleshoot Route Maps and PBR (It contains 3 trouble tickets 15.1.4.1-3).

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

8.1 Route Maps

A Route-Map is an ACL that provides a similar structure of top-down processing, sequence-number lines and permit or deny but uses more features such as Match and Set criteria (similar to the "if-then" logic in programming) which make the route-map a sophisticated ACL. The route-map tool has several uses, the most important of them are listed below (the main applications of route-map are in the control plane and deal with routes, only PBR uses the route map in the data plane) [1], [20]:

- Route filtering using route-map in distribute-list tool.
- Route filtering and manipulation when performing redistribution.
- Policy-Base Routing or PBR
- BGP

Route Maps consist of four components [1], [20]:

- **Sequence Number**: Route maps are configured with an ordered sequence number for each statement like ACLs and Prefix-Lists. If nothing is specified as a sequence number, this number is automatically increased to 10.
- **Processing Action**: Each statement of a route map (like ACLs and Prefix-Lists) is configured with a permit or deny a route as the Processing Action. The default value of Processing Action is Permit.
- **Conditional Matching criteria**: The match keyword is used, to select a specific prefix based on the prefix characteristics such as next-hop, BGP path attribute, and so on.
- **Optional Action**: In addition to selecting a specific prefix, the set keyword can be followed to modify, add, and remove the prefix attributes and characteristics.

The route map can be implemented with the following global configuration command [1], [20]:

```
#route-map route-map-name [permit | deny] [sequence-number]
 #match {as-path acl-number | community community-list| …….}
 #set {local-preference 0-4294967295 | origin [igp | incomplete] | ……}
```

8.2 Conditional Forwarding (PBR, Local PBR)

Policy-Based Routing (PBR) is a feature that puts network engineers in complete control of what routing doing. PBR provides conditional routing and forwarding of packets that break the rules of routing based on the available information in the routing table. PBR provides the routing capabilities based on src/des IP addresses, protocol types such as TCP, UDP, ICMP, etc [1],[5].

The following is an example of source-based IP routing using the PBR (Configuration commands are provided from [1]). All routers in Figure 8. 1 are configured with Single-Area OSPFv2. To get access to the network resources (192.168.1.0/24) from D1's LAN (Lo2) traffics follow the following path D1---> R1 --> R2 --> R3 --> D2, this is due to the best route provided by the R2 Ethernet link instead of the R3 serial link:

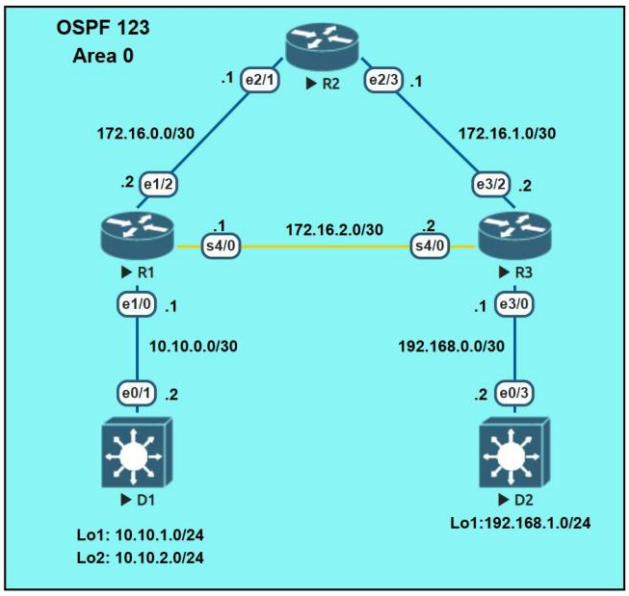


Figure 8. 1: Network topology for the emulated lab "Path Control Using PBR"

To pass traffic through the serial link between R1 and R3 routers, PBR is implemented on R1 based on the source IP address:

- Specify the source IP address by defining a standard ACL:

```
R1(config)#ip access-list standard Lo2-ACL
R1(config-std-nacl)#remark ACL marches D1 Lo2 traffic
R1(config-std-nacl)#permit 10.10.2.0 0.0.0.255
```

- Create a route map:

```
R1(config)#route-map R1-to-R3 permit
R1(config-route-map)#description RM to forward Lo2 traffic to
R3
```

- Using "Match" in the route map, select the prefix specified in the ACL *Lo2-ACL*:

```
R1(config-route-map)#match ip address Lo2-ACL
```

- Allows traffic to pass through the serial link between R1 and R3 routers using the "Set" action in the route map:

```
R1(config-route-map)#set ip next-hop 172.16.2.2
```

- Apply the route map to the inbound interface (The inbound traffic is examined for PBR processing):

```
R1(config)#int e1/0
R1(config-if)#ip policy route-map R1-to-R3
```

- Confirming the traffic path R1 --> R3 --> D2 by traceroute command:

```
D1#traceroute 192.168.1.1 source lo 2
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.0.1 1 msec 0 msec 0 msec
  2 172.16.2.2 5 msec 4 msec 3 msec
  3 192.168.0.2 5 msec *  6 msec
```

9. ROUTE REDISTRIBUTION

This chapter covers in more detail the laboratory material related to Route Redistribution. The following labs are dedicated to this topic:

- Lab **#16.1.2_** Configure Route Redistribution Between EIGRP and OSPF.
- Lab **#16.1.3_** Configure Route Redistribution Within the Same Interior Gateway Protocol.
- Lab **#16.1.4_** Configure Route Redistribution Using BGP.
- Lab **#17.1.2_** Troubleshoot Redistribution (It contains 3 trouble tickets 17.1.2.1-3).

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

Route Redistribution_ Injecting the routes (static, connected, or learned from a routing protocol or routing process) from one routing protocol (Source Protocol) to another routing protocol (Destination Protocol) is called Route Redistribution [1], [20]. In route redistribution, the presence of routes in the routing table is a prerequisite. Redistribution can occur between different routing protocols or different routing processes within the same routing protocol. The redistributed routes must have an understandable route metric (seed metric) into destination protocol [1], [20], [21].

Routes can be redistributed within the same IGP or between different routing protocols according to the following command [1]:

```
redistribute {connected | static | eigrp as-number | ospf process-id
[match {internal | external [1|2]}] | bgp as-number} [destination-
protocol-options]
```

9.1 Redistribution routes within the same IGP

In redistributing EIGRP (Static and directly connected)-to-EIGRP or OSPF-to-OSPF, the route metric values are maintained and no seed metric definition is required [1].

9.2 Redistribution routes between different routing protocols

The following describes the behavior of routing protocols from a destination perspective:

EIGRP redistribution: By default, EIGRP has the seed metric of infinity (unreachable). Route redistribution into EIGRP needs to define the seed metric for specific values of bandwidth, load, delay, and maximum transmission unit (MTU), otherwise, the routes

will not be installed into the EIGRP topology table but connected and static routes are an exception. By default, the redistributed routes in EIGRP have an AD of 170 and are identified by "D EX" in the routing table. The following command provides route redistribution into EIGRP: [1], [20], [21]

```
redistribute source-protocol [metric bandwidth delay reliability load
mtu] [route-map route-map-name]
```

OSPF redistribution: By default, external OSPF routes are classified as Type-2 (E2), AD of 110, and put a seed metric of 20 for all protocols except BGP-sourced routes which is 1 [21]. The main difference between E1 and E2 is that the E2 metric is the same for all hops after ASBR, while the E1 metric increases hop by hop and it is more preferred over E2 [1].

```
redistribute source-protocol [subnets] [metric metric] [metric-type {1
| 2}] [tag 0-4294967295] [route-map route-map-name]
```

[**subnets**]: This keyword enables redistribution for all classful and classless prefixes.

Redistributed routes into a normal area of OSPFv2 are advertised to another area as Type 5 LSA this is while NSSA or Totally NSSA areas advertise as Type 7 LSA [1].

BGP redistribution: By default, only eBGP routes are redistributed into IGP, and certainly not to forget BGP ability to manage the large routing table. Route redistribution into BGP doesn't require a seed metric because BGP is a Path Vector routing protocol. The following BGP attributes are assigned for the redistributed routes into BGP [1]:

- The Origin is shown incomplete (?) in the BGP table.
- The Weight attribute is changed from 0 to 32768.
- The source protocol IP address is set as the Next-Hop address.
- The MED value is set for the path metric value of the source protocol.

By default, due to loop prevention rules, iBGP routes are not allowed to redistribute into any IGP. To redistribute the iBGP routes into IGP it is needed to configure the following command within a BGP process [1]:

```
(config)#router bgp <AS#>
(config-router)#address-family [ipv4|ipv6]
(config-router-af)#bgp redistribute-internal
```

By default, it is only intra-area and inter-area OSPF routes are redistributed into BGP. The following command provides the redistribution of the OSPF external route in BGP [22], [1]:

```
redistribute ospf process-id match internal external 1 external 2
```

9.3 Troubleshooting

This section covers problems you may encounter when redistributing the route and the troubleshooting techniques.

Redistribution means injecting routes from a source routing table into a destination's data structure (such as OSPF LSDB).
Two common problems that may arise are Suboptimal routing and Routing loops when redistributing at multipoint between two routing protocols [1]. In the case of suboptimal routing troubleshooting, determine the link’s speed, identify the seed metric used, and if necessary, modify the seed metric [1]. In the case of loop routing, first of all consider that the internal route is more preferable to the external route with lower AD, this default value of AD can be changed using the following command [1]:

```
EIGRP: #distance eigrp ad-internal ad-external
OSPF:  #distance ospf {external | inter-area | intra-area} ad
BGP:   #distance bgp external-ad internal-ad local-routes
```

The second point to consider to avoid the problem of the routing loop is the redistribution of the route from the destination back into the source. This problem can be solved by using the route tag, which is accomplished with route maps [1].

The following are the most useful commands for troubleshooting issues related to route redistribution [1]:

```
show ip route ip-address (examine a redistributed route in the routing table)
show ip protocols (verifies which protocols are being redistributed)
show ip eigrp topology
show ip ospf database
show ip bgp
```

10. VRF-LITE, GRE TUNNELS

This chapter covers in more detail the laboratory material related to VRF-Lite and GRE Tunnels. The following labs are dedicated to these topics:

- Lab **#18.1.2_** Implement VRF-Lite.
- Lab **#19.1.3_** Implement a GRE Tunnel.

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

10.1 VRF-Lite

VRF is a technology for dividing or separating a physical router into different virtual routers with isolated routing and forwarding tables. VRF without MPLS is called VRF-Lite [1], [23]. The automatic separation of traffic with VRF ensures network security and eliminates the need for data authentication and encryption. VRF is created by defining a name as the first step, then the intended interfaces must be assigned to the created VRF [1]:

```
(config)#vrf definition <vrf-name>

(config)#interface <interface-type> <interface-number>
(config-if)#vrf forwarding <vrf-name>
```

Because each VRF instance has its routing table, interfaces in different VRF can be configured with the same IP address without overlapping each other [23]. The following command verifies the assigned interface for each specific VRF [1]:

```
show ip vrf interfaces
```

10.2 GRE Tunnels

Generic Routing Encapsulation (GRE) is a protocol for tunneling a variety of protocols over an IP platform. To allow packets to be exchanged between the endpoints of the tunnel, the router encapsulates the packets with a new header (GRE IP header) that contains the destination IP address. The router at the destination side obtains the original packet by de-capsulation. This protocol is a simple tunneling technique that can encapsulate any data and send it to the physical interfaces of the router [1], [7].

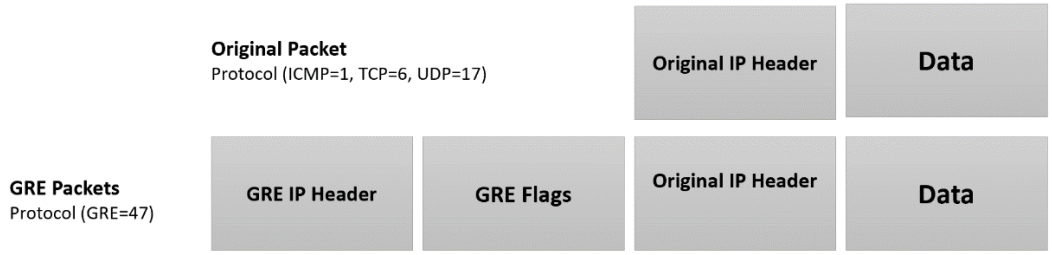


Figure 10. 1: IP packet before and after encapsulation with a GRE header

The following are the configuration steps for a GRE tunnel [1], [7]:

```
#interface tunnel tunnel-number
#ip address ip-address subnet-mask

In case of configuring a tunnel for IPv6:
#tunnel mode gre ipv6

Tunnel source can be a loopback or a physical interface:
#tunnel source {ip-address | interface-id}

Tunnel destination is the remote router's underlay IP address:
#tunnel destination ip-address

A reference bandwidth should be configured on the virtual interface
regarding best-path calculation (EIGRP):
#bandwidth <#>
```

11. DMVPN

This chapter covers in more detail the laboratory material related to DMVPN services and their implementations. The following implementation labs are dedicated to the topics of this chapter:

- Lab **#19.1.3**_Implement a DMVPN Phase I
- Lab **#19.1.4**_Implement a DMVPN Phase III
- Lab **#19.1.5**_Implement an IPv6 DMVPN Phase III
- Lab **#20.1.2**_Configure Secure DMVPN Tunnels

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

DMVPN_ To connect a company's branches, tunnels can be created between the company's branches across the public network, such as the internet. If the number of branches is large, creating tunnels manually will be very difficult and time-consuming, as well as requiring a fixed IP address and reconfiguration (in an IPsec-based VPN) for each newly installed branch. One way to overcome these problems is using the Dynamic Multipoint Virtual Private Network (DMVPN).
DMVPN consists of the Hub-Spoke architecture. Usually, the router located at the central office is considered a Hub, and the routers at the other branches are considered Spokes [5]. DMVPN is supposed to be a fully dynamic solution, with a routing protocol (IGP or BGP, but not static routing) implemented on top of the DMVPN cloud to exchange information about individual sites connected to Spokes as well as the Hub.
The DMVPN uses technologies such as Multipoint GRE (mGRE), Next Hop Resolution Protocol (NHRP), and IP security (IPsec) to create its overlay network [7].
DMVPN also solves the sub-optimal routing problem. This problem occurs when two spoke routers want to communicate with each other through a hub. In DMVPN phases II and III, on-demand Spoke-to-Spoke tunnels can be implemented [1].

DMVPN elements:

The following 4 main elements must be considered to set up a DMVPN network:

- **Routing**: Routing protocols are used to build Hub-Spoke adjacencies (underlay and overlay networks). When using a routing protocol on a DMVPN network, two elements need to be considered: a) Split Horizon: It prevents route advertisement from being sent back through the interface it received on, b) The Next Hop address: It can be the Hub IP address itself or the Spoke IP addresses.

EIGRP is the recommended routing protocol for DMVPN, due to its suitability for Hub and Spoke designs. It can disable split Horizon with the no ip split-horizon eigrp command, the Next-Hop address can be changed with the no ip next-hop-self eigrp command, and EIGRP stub router feature [1], [37].

In larger deployments, BGP is the most useful routing protocol, it can be implemented by eBGP or iBGP. eBGP ensures that the Next-Hop address changes automatically when sending an update, but the limitation of this solution is that, Spokes must be configured in AS different from Hub that may require 32-bit AS numbers in larger deployments. It can be found a better solution with iBGP, all DMVPN devices can be deployed in one AS, the Hub router acts as a Route-Reflector, and Next-Hop must be changed on Spokes with neighbor next-hop-self all command [1], [37].

- **mGRE**: Multipoint GRE (GRE) is a type of GRE tunnel that provides the ability to have multiple GRE tunnels on one interface [5]. It is useful when we want to connect two Spokes directly without traffic passing through the Hub. Like the GRE, mGRE can transmit traffic through various protocols. The mGRE uses NHRP to find the destination IP address of the tunnels [5].

- **NHRP**: Next Hop Resolution Protocol (NHRP) is an ARP-like protocol, that provides address resolution [1]. In the mGRE tunnel, the underlay destination address of the tunnel is not known, because it is not a typical point-to-point GRE tunnel. NHRP is used to do this dynamically by mapping overlay IP address to underlay IP address.

  NHRP works on a client-server model, Hub router acts as a Next Hop Server (NHS) while the Spoke router acts as a Next Hop Client (NHC) [7]. Spoke is statically configured with the IP address of the Hub, then it can report its IP addresses (underlay and overlay) to the NHS.

  In the case of Spoke-to-Spoke communication, the spokes are configured with mGRE interface and registered their IP addresses (underlay and overlay) with the Hub, then if one of the Spoke intends to communicate with another one, it will ask the Hub about the underlay address of that Spoke, and the Hub will provide this information to them by referring to its table, to provide direct communication between the spokes.

  NHRP works by exchanging NHRP messages. The following Table 11. 1 is a list of NHRP messages [1]:

Message Type	Description
Registration	This message is used by Spokes to inform Hub about their underlay (Public) address.
Resolution	It consists of two messages: Request and Reply. Spoke1 asks Hub for Spoke2 public address in a Request message, Hub provides Spoke2 address in a Reply message.
Redirect	In DMVPN Phase III, Hub uses this message to inform an optimal path (spoke-to-spoke tunnel) is available.

Table 11. 1: Types of NHRP messages

- **IPsec**: DMVPN itself doesn't provide a protected communication [1]. The DMVPN at the high level can be described as a group of GRE tunnels, that make up the DMVPN network, since the GRE itself is a clear-text protocol, meaning that the DMVPN itself does not provide any types of security services. Security services such as encryption, integrity checking, and authentication are accomplished using another protocol often referred to as IP security (IPsec) [5].

DMVPN was released in three phases, each phase built on the previous one with additional functions:

11.1 DMVPN PHASE I

In DMVPN Phase I, the communication between spokes is done through the HUB. The main advantages of this method are **a)** Reduction in the number of GRE tunnels at the Hub side, **b)** Flexible Spoke configuration. DMVPN Phase I doesn't support Spoke-to-Spoke tunnels and redundancy. It uses mGRE but only on the Hub router, because of the lack of mGRE configuration on Spokes, there is no direct communication between Spokes [1].

11.1.1 Implementing DMVPN Phase I

The implementation consists of two parts: Hub and Spoke configurations. All configuration examples are based on network topology in Figure 11. 1:

Hub Configuration:

In Figure 11. 1 R1 is intended as a Hub router and must be configured with the following commands [1] [38]:

- Create the tunnel:

```
R1(config)#interface tunnel 1
```

- Identify the IP address of the tunnel (Overlay network):

```
R1(config-if)#ip address 100.100.100.1 255.255.255.248
```

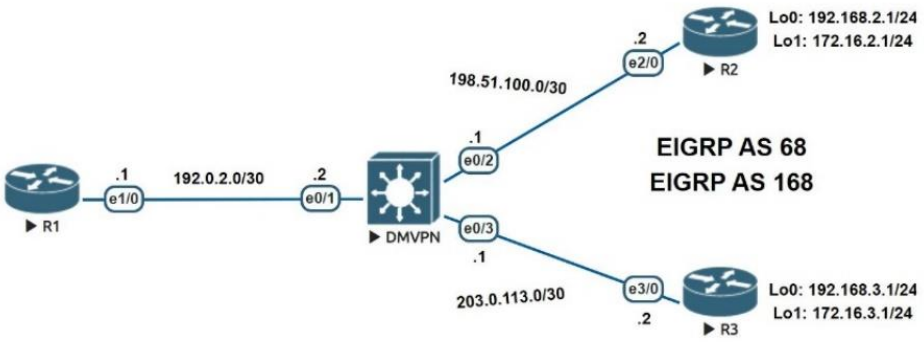


Figure 11. 1: Network topology for the emulated DMVPN labs

- Specify interface e1/0 as the tunnel source (Encapsulating Interface):

```
R1(config-if)#tunnel source e1/0
```

- Specify the tunnel mode as an mGRE tunnel (multipoint interface), because the Hub is connected to multiple Spokes. In the mGRE implementation, no tunnel destination (NMMA or underlay address) will be manually specified and the Hub will learn about Spoke by the NHRP:

```
R1(config-if)#tunnel mode gre multipoint
```

- Enabling multipoint tunnels on a single interface requires a tunnel key, this is optional but must match between the two routers when configured:

```
R1(config-if)#tunnel key 999
```

- In this part of the configuration, R1 will be configured as an NHRP server (NHS). You can start by specifying the network ID, which must be the same between Hub and Spokes:

```
R1(config-if)#ip nhrp network-id 1
```

- NHRP network security with *ip nhrp authentication* command:

```
R1(config-if)#ip nhrp authentication NHRPauth
```

- The following command creates the NHRP table and enables R1 to add Spokes to the NHRP table which enables the use of the dynamic routing protocols between the Hub and Spokes. This command specifies the destination of multicast packets because the IP address for Spokes is not specified, so the dynamic keyword must be used on the Hub to automatically add Spoke's IP address to the list of multicast destinations when they first register in the Hub:

```
R1(config-if)#ip nhrp map multicast dynamic
```

- Determine the interface bandwidth for the proper operation of routing protocols (EIGRP is used in this example):

```
R1(config-if)#bandwidth 4000
```

- Determine the Maximum Transmission Unit (MTU) value:

```
R1(config-if)#ip mtu 1400
```

- Specifying the TCP Maximum Segment Size (MSS) which ensuring router will edit the payload of TCP three-way handshake in case of exceeding the MSS:

```
R1(config-if)# ip tcp adjust-mss 1360
```

Spoke Configuration:

In Figure 11. 1 R2 and R3 routers are considered as Spoke routers and must be configured with the following commands [1], [38]:

- Creating tunnel, Identifying its IP address and source:

```
R2(config)#interface tunnel 1
R2(config-if)#ip address 100.100.100.2 255.255.255.24
R2(config-if)#tunnel source loopback 0
```

- Specify the IP address of interface e1/0 (R1) as the tunnel destination (point-to-point GRE interface is configured on Spoke):

```
R2(config-if)#tunnel destination 192.0.2.1

R2(config-if)#tunnel key 999
```

- In this part of the configuration, R2 will be configured as an NHRP client (NHC). It can be started by specifying the network ID that must be the same as configured on NHS for this DMVPN network:

```
R2(config-if)#ip nhrp network-id 1
R2(config-if)#ip nhrp authentication NHRPauth
```

- Specify the NHS address (overlay network) for the NHC:

```
R2(config-if)#ip nhrp nhs 100.100.100.1
```

- Static mapping for the Hub overlay address and its underlay address (How to get to the Hub):

```
R2(config-if)#ip nhrp map 100.100.100.1 192.0.2.1
```

- Static mapping the destination address (NBMA) for sending multicast packets:

```
R2(config-if)#ip nhrp map multicast 192.0.2.1
```

```
R2(config-if)#ip mtu 1400
```

```
R2(config-if)#ip tcp adjust-mss 1360
```

In this example, both underlay and overlay networks are implemented by the EIGRP routing protocol, the DMVPN switch also must be enabled with EIGRP for the Underlay network.

The following are the verification commands for the DMVPN and NHRP status respectively:

```
show dmvpn detail
show ip nhrp detail
```

11.2 DMVPN PHASE II

The main difference with DMVPN Phase II is the ability to create direct Spoke-Spoke tunnels on the same DMVPN network, and allowing for redundancy. The main requirement for DMVPN Phase II is that all routers must be configured with mGRE interfaces, another key change would be Next-hop, which should point to Spoke. In addition to the NHRP Registration message, DMVPN Phase II uses two other important messages: **1)** Resolution Request (Spoke asks the Hub for the NBMA of the Next-Hop) **2)** Resolution Reply (Sent out in response of Resolution Request) [1].

11.3 DMVPN PHASE III

DMVPN Phase 3 establishes the communication between two Spoke within the same or different DMVPN networks (Multilevel hierarchical DMVPN). DMVPN Phase III supports effective route Summarization where the Next-Hop has been pointed to the Hub. In addition to the NHRP Registration and Resolution messages, the NHS in DMVPN Phase III uses another message known as the Redirect (ip nhrp redirect) message. When the Hub sends an NHRP Redirect message, Spokes use shorter paths with the NHRP shortcut switch feature (ip nhrp shortcut), which allows the NHRP cache to overwrite Cisco Express Forwarding (CEF) [1].

11.3.1 Implementing DMVPN Phase II

The implementation consists of two parts: Hub and Spoke configurations. All configuration examples are based on network topology in Figure 11. 1 [1]:

Hub Configuration:

The configuration on the DMVPN Hub has the following additional command compared to the previous DMVPN Phase I configuration:

- Informing the Spokes that they can have direct reachability between each other:

```
R1(config)#interface tunnel 1
```

```
R1(config-if)#ip nhrp redirect
```

Spoke Configuration:

There are two additional commands on Spoke routers compared to the previous Phase I DMVPN configuration:

- To communicate with all DMVPN network devices, Spokes must be configured with multi-point interfaces (mGRE):

```
R2(config)#int tunnel 1
R2(config-if)# no tunnel destination
R2(config-if)#tunnel mode gre multipoint
```

- Spokes can overwrite their CEF table when they receive a Redirect message from the hub:

```
R2(config-if)#ip nhrp shortcut
R2(config-if)#ip nhrp map multicast dynamic
```

11.4 DMVPN and IPv6

DMVPN has fully supported IPv6 both underlay and overlay DMVPN networks. DMVPN works by using the GRE protocol for encapsulation, where GRE itself is a multi-protocol tunneling technique that allows not only to carry IPv4 but also IPv6 and other packet types. In DMVPN for IPv6, NHRP mappings can be a mix of IPv4 with IPv6 or separate for each of them [1].

11.4.1 Implementing IPv6 DMVPN

Regarding the implementation, the same commands are used in implementing DMVPN for IPv6 but with the following differences [1]:

- Replace IP keyword with IPv6
- Specifying the tunnel mode (mGRE) for IPv6:

```
tunnel mode gre multipoint ipv6
```

- Specifying the NHS address (overlay network) and mapping the NHS underlay address to its overlay address:

```
ipv6 nhrp nhs <ip-address> nbma <ip-address> multicast
```

11.5 Securing DMVPN Tunnels

This section covers two topics of IPsec Fundamentals and IPsec over DMVPN:

11.5.1 IPsec fundamentals

IP Security (IPsec) is one of the most common implementations of VPNs specified in RFC 2401 "Internet Protocol Security Architecture". IPsec VPNs are built on Layer 3. The main reason for using IPsec is usually the ability to provide security in communications by the following security services [1], [5]:

- **Authentication**: Ensure that the session is built with the correct communication partner. It is accomplished by a Pre-Shared Key (statically) or Certificate (dynamically).
- **Confidentiality**: Hiding information from any kind of party by using encryption algorithms by the sender.
- **Data Integrity**: Ensure that the messages received integrally by the destination communication partner which has not been changed during the transmission. It is accomplished by hashing algorithms.
- **Anti-reply protection**: Avoid injecting old packets into the current session.
- **Periodic rekey**: Creating new security keys in different time interval

IPsec is not a single protocol; in fact, it is a framework and consists of multiple standards and protocols that each of them has a certain function. The IPsec architecture consists of the following independent components [1], [2]:

- **Security Protocol:** It includes two protocols for keeping information confidential and creating data integrity: **1)** Authentication Header or AH (**RFC 4302**) and **2)** Encapsulation Security Payload or ESP (**RFC 4303**). The AH protocol offers authentication, integrity, and reply protection. It prevents data from changing over the Internet by creating a digital signature. The ESP offers authentication, integrity, reply protection, and confidentiality. ESP and AH support both modes of encapsulation (Transport and Tunnel).
- **Key Management:** The process of creating, sending, and storing the key is called Key management. The key is used to secure the communication by encrypting and decrypting the data. The most two important protocols related to Key management in the control plan are **a)** Internet Security Association & Key Management Protocol (ISAKMP) and **b)** Internet Key Exchange (IKE). By default, IPsec uses the IKE which includes two versions of IKEv1 and IKEv2.
- **Security Associations (SA):** Security associations specify security parameters such as protocols, algorithms, and keying materials that must be agreed upon and matched between two IPsec endpoints. There are two types of SA, **a)** IPsec SA_used for data plane functions to secure the data transmission, at least one IPsec SAs must be established for each incoming and outgoing traffic, **b)** IKE SA_use for control plane functions such as management of IPsec SA and IPsec key management, only One IKE SA is established between two endpoints.

IPsec in the Data Plane can run using two encapsulation modes (ESM modes) [1]: **1)** Tunnel mode_ The entire original packet is encrypted and a new set of IPsec headers is added to be used to route packets and provide overlay functions and **2)** Transport mode_ Only the packet payload is encrypted and the original IP headers are used to route the packets. In ESP, the payload is part of the main packet encapsulated in the IPsec headers, not the actual data without any headers. [1].
IPsec VPNs are negotiated in two phases: **1)** Phase I which is performed in one of the two modes of Main Mode (MM) or Aggressive Mode (AM), **2)** ISAKMP/IKE Phase II which is performed in the same mode of Quick Mode (QM). In case of successful Phase I negotiation, an IKE Security Association (SA) and in case of Phase II successful negotiation, results in two separate IPsec SAs for inbound and outbound communication. In case of successful Phase I negotiation, an IKE Security Association (SA) and in case of Phase II successful negotiation, results in two separate IPsec SAs for inbound and outbound communication [2], [54].

11.5.2 IPsec over DMVPN

It is mentioned at the beginning of this chapter that DMVPN itself is not a protected communication medium. All the devices within a DMVPN network need to be configured with the same settings otherwise no tunnel will be established. The following IPsec deployment is implemented on R1 according to the network topology in Figure 11. 1 [1]:

First Step (Create the IKE policy): Define the four main elements for negotiation (Authentication, Encryption algorithm, Hash algorithm, and DH group) [53]:

```
R1(config)#crypto isakmp policy 99
R1(config-isakmp)#hash sha384
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#group 14
R1(config-isakmp)#authentication pre-share
```

Second Step: Since the Pre-shared key has been selected as the authentication method, it is necessary to configure the key (password) and specify the IP address of the peer, which is 0.0.0.0 in this example (to match multiple peers):

```
R1(config)#crypto isakmp key DMVPN@key# address 0.0.0.0
```

Third Step (Create IPsec transform set): IPsec transform set specifies one of the security protocols (AH or ESP) for encrypting data (ESP is the most appropriate protocol because it provides data confidentiality) and one of the two encapsulation modes (Tunnel vs Transport) [1]:

```
R1(config)#crypto ipsec transform-set DMVPN_TRANS esp-aes 256 esp-
sha384-hmac
R1(cfg-crypto-trans)#mode transport
```

Fourth Step (Create the IPsec Profile): IPsec protection is enabled through an IPsec Profile:

```
R1(config)#crypto ipsec profile DMVPN_PROFILE
R1(ipsec-profile)#set transform-set DMVPN_TRANS
```

Fifth Step: Applying the IPsec Profile on the tunnel interface:

```
R1(config)#interface tunnel 1
R1(config-if)#tunnel protection ipsec profile DMVPN_PROFILE
```

12. ACL AND PREFIX LIST

This chapter explains two topics of Access Control List (ACL) and Prefix List. Three Troubleshoot labs are dedicated to the topics of this chapter:

- Lab **#21.1.2**_Troubleshoot IPv4 ACLs (It contains 3 trouble tickets 21.1.2.1-3).
- Lab **#21.1.3**_Troubleshoot IPv6 ACLs (It contains 3 trouble tickets 21.1.3.1-3).
- Lab **#21.1.4**_Troubleshoot Prefix Lists (It contains 2 trouble tickets 21.1.4.1-2).

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

12.1 Access Control Lists (ACLs)

An ACL is composed of one or more Access Control Entries (ACEs), each ACE is assigned a sequence number with a list of 'permit' or 'deny' packet classification which is processed in sequential order until the information within the packet match against one of the ACEs [7], [1]. In the case of non-matching, all the packets will be denied due to the presence of an implicit deny ACE at the end of each ACL [1]. Access Control List (ACL) is used to match traffic (in NAT, or QoS) or control traffic (in route filtering, route redistribution, or policy-based routing) [1]. The created ACLs must be applied to an interface, service, or feature otherwise it is useless [1].
In this chapter, only two types of ACLs are compared, which are standard and extended ACLs, each of them can be identified and referenced by two styles, Number or Name. When an ACL is referenced by a name, the type of ACL should be specified before the name, and compared to the number style, the name style allows you to modify individual ACEs [9].

12.1.1 Standard ACLs

This type of ACL matches/controls the traffic based on the source IP address of the packet [7]. It can be created in the range number of 1-99 or 130-1999 [1]:

Configure Standard Number ACL

```
Router(config)#access-list acl-num { deny | permit } source [source-wildcard]
```

Configure Standard Name ACL

```
Router(config)# ip access-list standard acl-name
Router(config-std-acl)# {deny | permit} source [source-wildcard]
```

12.1.2 Extended ACLs

This type of Access-List matches/controls the traffic based on the protocol type, source/destination IP address, source/destination TCP/UDP ports, or a message type for ICMP which makes a powerful tool [7], [9]. If the filtering is based on an IP address called the Host level filtering (generic filtering) and filtering based on the Layer 4 protocols such as TCP/UDP headers or ICMP called the Application -level filtering (more specific filtering) [9]. The ACL number can be in the range of 100-199 or 2000-2699 [1], [9]. The following command provides the implementation of Extended ACLs:

Configure Extended Number ACL

```
Router(config)#access-list acl-num permit|deny IP_protocol
source_address  source_wildcard_mask [operator
source_port_#|Application-name] destination_address
destination_wildcard_mask [operator destination_port_#|Application-
name] [icmp_message]
```

Configure Extended Name ACL

```
Router(config)# ip access-list extended acl-name
Router(config-ext-acl)# permit|deny IP_protocol source_IP_address
wildcard_mask [operator source_port_#|Application-name]
destination_IP_address wildcard_mask [operator
destination_port_#|Application-name] [icmp_message]
```

IP_protocol keyword: It is used to determine the level of filtering, Host or Application

To use ACL for filtering, ACL should be applied on the interface with the following command [1]:

IPv4 ACL for filtering

```
ip access-group {acl-number | acl-name} {in | out}
```

IPv6 ACL for filtering

```
ipv6 traffic-filter {acl-name} {in | out}
```

12.2 Prefix Lists

Prefix-list is similar to Access-list except they can match the routes based on a subnet mask, so a Prefix List has more granular control for route filtering [1].
A prefix-list with the three keywords of eq (equal to), ge (greater or equal to), and le (less than or equal to) will have flexibility in identifying routes based on a subnet mask. In case of having an exact match of the network, No eq or No les is used. Prefix-list can be created as per the following command [1], [25]:

```
ip prefix-list list-name [seq seq-value] {permit|deny} prefix [eq length|[ge length] [le length]]
```

12.3 Troubleshooting

To troubleshoot the ACL/Prefix-List network problems, that should get understand how ACLs/Prefix-Lists behave, the following are provided from [1]:

- Top-down processing: An ACL/Prefix-List composed of various entries/ sequences which have a top-down (lowest-highest) processing.
- Straight implementation after the first matching of entry.
- (IPv6 ACL) Implicit permit icmp nd-na/ns: Permit these two types of messages, A) Neighbor Advertisement (NA) and B) Neighbor Solicitation (NS) to determine the associated layer 2 address with an IPv6 address:

```
Permit icmp any any nd-na
Permit icmp any any nd-ns
```

- Implicit deny any: This is an invisible entry that denies the packet if nothing to match in top entries.
- The created ACL can be checked with the following command:

```
#show access-lists
```

13. DEVICE ACCESS & FILE MANAGEMENT

This chapter deals with two topics of Device Access and File Management on CISCO devices, one Troubleshoot lab is dedicated to the topics of this chapter:

- Lab **#23.1.2**_Troubleshoot Device Access and File Transfer (It contains 3 trouble tickets 23.1.2.1-3).

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

13.1 Device Access

This section covers the different methods and protocols for accessing the device CLI:

13.1.1 Terminal Lines

There are three different ways (Terminal Lines) to access the IOS device CLI [1]:

- **Console port** (cty line): This port provides local access to the device. It is shown as line con 0 in the running/startup configurations and the output of the show line command shows as cty.
- **Virtual terminal** (vty line): These lines are logical and provide remote Telnet/SSH access to the device. By default, it is shown as line vty 0 4 in the configuration.
- **Auxiliary port** (aux line): This port provides remote access to the device via a modem (Out of Band). It is shown as line aux 0 in the configuration.

13.1.2 Telnet

Telnet is a protocol that provides a TCP / IP remote connection to the device using vty lines. This is not recommended by Cisco as it supports plaintext communication [1].

13.1.3 Secure Shell (SSH)

Secure Shell Protocol (SSH) provides an encrypted connection for remote access to the device compared to Telnet [1]. It is recommended to use SSH protocol instead of Telnet due to the secured session, it can be enabled according to the following steps [27]:

```
Both hostname and Domain-name are needed to generate the encryption keys:
Router(config)#hostname host-name
host-name(config)#ip domain-name domain-name

Allow SSH client access by specifying a username:
host-name(config)#username username password password
```

```
Generate the encryption keys <specify the size of the key modulus>:
host-name(config)# crypto key generate rsa <360-4096 >

Specify the SSH protocol version:
host-name(config)#ip ssh version version-value

The local database must be used for password, and enable ssh protocol:
host-name(config)#line vty 0 15
host-name(config-line)#login local
host-name(config-line)#transport input ssh
```

13.2 File Management

Today, almost all enterprise networks use protocols such as FTP, TFTP, SCP, HTTP, and HTTPS to remotely manage and transfer device files. This includes upgrading IOS, managing the device file system, restoring configurations, and performing backups [1].

13.2.1 Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol (TFTP) is an insecure file transfer protocol, there is no authentication or encryption method. TFTP only uses port 69 UDP to transfer files from the TFTP client to the TFTP server and vice versa [1]. It is possible to use a Cisco router as a TFTP server by enabling this feature with the tftp-server global configuration command. This feature has limitations, it can only be used to download files. To have a suitable and functional TFTP server for receiving and sending files, it is recommended to install and run a TFTP server software on a Windows / Linux server or host [1], [26].
The following command is used to transfer files between a TFTP Client and a TFTP Server [1]:

```
#copy source destination
```

13.3 Troubleshooting

This section provides tips for troubleshooting network issues related to the topics in this chapter. In addition, trouble ticket #23.1.2.1 from “Troubleshoot labs” has been worked at the end of this section, on behalf of other trouble tickets.

Telnet:

- Ensuring there is a connection to the remote device.
- The vty line should ask the user for a username/password that is configured locally on the device or uses an AAA server.
- Ensuring there is no ACL configured to block source stations or port number 23.

SSH:

- The same version of SSH v1/2 should be used on both sides.
- Ensuring that SSHv2 is configured with an RSA key size greater than 768.
- Ensuring that there is no ACL configured to block source stations or port number 22.

TFTP:

The following include Cisco recommendations for troubleshooting TFTP file transfer problems [1]:

- Make sure there is a connection between the TFTP Client and the Server.
- Make sure both the client and the server have enough storage space.
- Make sure there is no ACL configured to block TFTP traffic from source to destination.
- In case of using a management interface for TFTP traffic, the management interface should be specified as the source interface for TFTP traffic, using the following command line:

```
ip tftp source-interface interface_type interface_number
```

13.3.1 Trouble ticket #23.1.2.1 (File transferring by using TFTP)

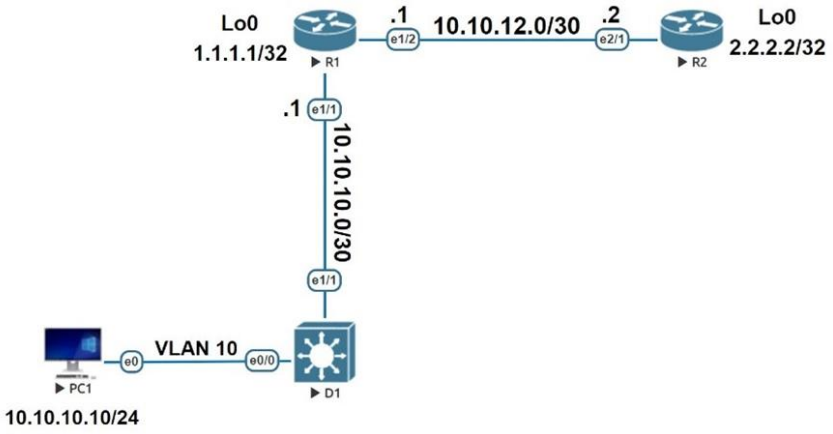


Figure 13. 1: Network topology for the emulated lab "Troubleshoot Device Access and File Transfer"

All devices in this lab configured with access and file transfer capabilities, SolarWinds TFTP server is installed on PC1. This ticket is related to a problem transferring files using the TFTP protocol, R2 configuration file can't be copied to the TFTP server:

```
R2#copy running-config tftp://10.10.10.10/r2-config.txt
Address or name of remote host [10.10.10.10]?
Destination filename [r2-config.txt]?
…..
%Error opening tftp://10.10.10.10/r2-config.txt (Timed out
```

This problem between the client and the server can be due to a layer 1 issue, high memory utilization, or in most cases a bad ACL configuration that blocks TFTP traffic from source to destination or vice versa.
We assume that there is no lack of memory storage on both sides and the connection is checked, the server is reachable from the Client side.

```
R2#ping 10.10.10.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
```

As per Cisco recommendation, the network may be configured with an incorrect ACL, R1 in this network is configured with two ACLs of PERMIT-ADMIN and ALLOW-TFTP:

```
R1#sh access-lists
Standard IP access list PERIT-ADMIN
    10 permit 10.10.10.0, wildcard bits 0.0.0.255
Extended IP access list ALLOW-TFTP
    10 permit udp 10.10.0.0 0.0.255.255 eq tftp host 10.10.10.10
    20 permit udp 10.10.0.0 0.0.255.255 host 10.10.10.10 gt 1024
    …
    70 permit icmp any any
```

As expected, ACE #10 on R1 is incorrectly configured for TFTP traffic. According to Cisco's recommendation, it should be configured with the following command. The following configured ACE allows all TFTP traffic from the range of 10.10.0.0/16 network to get access into PC1 (10.10.10.10):

```
R1(config)#ip access-list extended ALLOW-TFTP
R1(config-ext-nacl)#no 10
R1(config-ext-nacl)#permit udp 10.10.0.0 0.0.255.255 host 10.10.10.10
eq tftp

The issue is resolved:
R2#copy running-config tftp://10.10.10.10/r2-config.txt
Address or name of remote host [10.10.10.10]?
Destination filename [r2-config.txt]?
!!
1968 bytes copied in 0.128 secs (15375 bytes/sec)
```

The following figures confirm a proper and functional TFTP server:

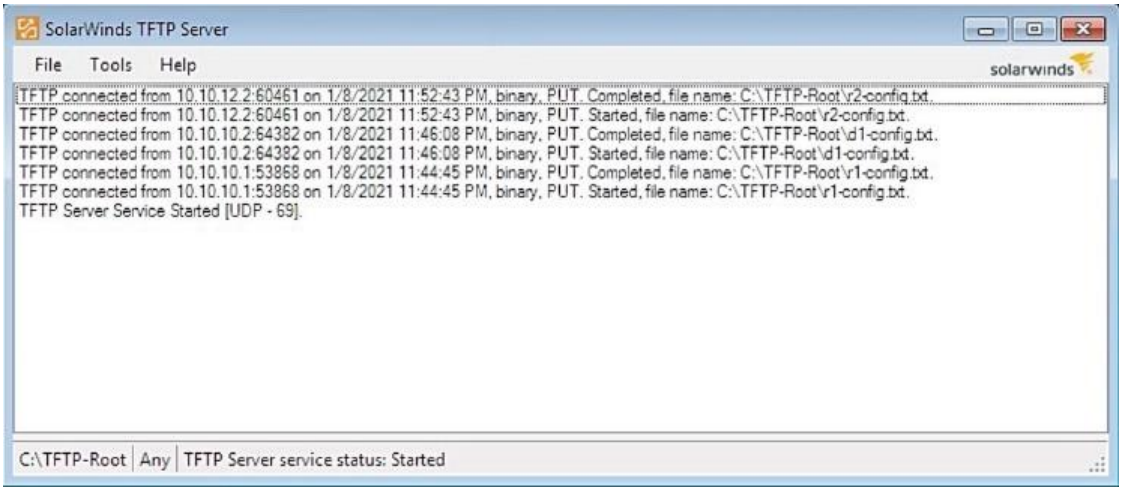


Figure 13. 2:The TFTP Server logs

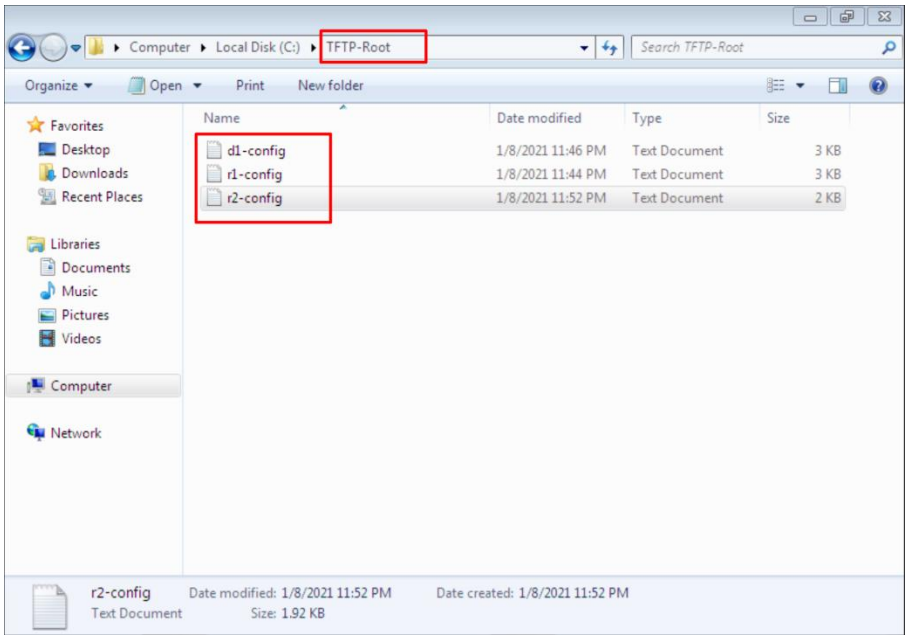


Figure 13. 3: The Copy of config files on PC1

14. INFRASTRUCTURE SECURITY

This chapter covers three network security protocols, including AAA, uRPF, and CoPP, three Troubleshoot labs are dedicated to the topics of this chapter:

- Lab **#22.1.2**_Troubleshoot IOS AAA (It contains 2 trouble tickets 22.1.2.1-2).
- Lab **#22.1.3**_Troubleshoot uRPF (It contains 1 trouble ticket 22.1.3.1).
- Lab **#22.1.4**_Troubleshoot CoPP (It contains 2 trouble tickets 22.1.4.1-2).

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

14.1 Authentication, Authorization, and Accounting (AAA)

AAA is a security framework for enabling three different security functions: **1)** Authentication_this is the process of claiming identity and then proving that identity (Username/Password), **2)** Authorization_this is a process for determining the level of access that is given after successful authentication, and **3)** Accounting_this is a process to explain how to track and record activities [1].

There are two main methods to deploy the AAA solutions, the first method is direct deployment on local network devices such as routers, switches, firewalls called Local AAA Authentication. This method is not scalable and it is limited in AAA functionality. The second method is called Server-Based AAA Authentication, it focuses on deploying AAA using an AAA server, which is based on a Client-Server model that each side communicates with each other using Remote Authentication Dial-In User Service (RADIUS) or Terminal Access Controller Access-Control System Plus (TACACS+) protocols [7].

The RADIUS protocol is an open IETF standard that encrypts only passwords, while the TACACS + protocol is a Cisco proprietary protocol that encrypts the entire message [5]. In Cisco , RADIUS uses UDP port 1645 for Authentication and Authorization , and 1646 for Accounting , this is while port numbers 1812 for authentication and 1813 for accounting are open standard RADIUS ports . The TACACS+ protocol uses TCP port 49 [1].

AAA is used in two ways: **1)** Network access control, **2)** Secure network access control [1]. TACACS+ is mainly used in network access or device administration, this is because of the two features that the protocol supports: **1)** Separates each of Authentication, Authorization, and Accounting but Radius combines Authentication with Authorization. **2)** Authorization parameters can be requested separately at different times. On the other hand, all authorization parameters are returned in a single response using a Radius server [1].

Radius is used in secure network access; this is because Radius supports Extensible Authentication Protocol (EAP). This is possible by using EAP to identify the user before accessing the network [1].

14.2 Unicast Reverse Path Forwarding (uRPF)

Unicast Reverse Path Forwarding (uRPF) is a network security solution in network routing to increase security and prevent IP Spoofing [1]. Generally, each router receives a packet and then sends it to the destination address according to the routing table through the appropriate interface regardless of the source IP. This is considered a weakness for network security because a malicious hacker using IP Spoofing technique can forge the source IP address [39] and route the packet that should normally be dropped through the router. The uRPF technique is security to deal with IP Spoofing attacks which increase security and prevent unauthorized use of resources.

The uRPF is implemented in three general modes in Cisco routers [1], [39]:

- **Strict**: In this mode of uRPF implementation, routers check the IP address of the source of the packet and ingress interface. If the source is reachable through the ingress interface in the routing table, the packet is allowed to be routed.
- **Loose**: In this mode, routers only check the source IP address of the packet. If the source address is reachable through any interface in the routing table (not a default route), the packet is allowed to be routed.
- **VRF**: In this type of uRPF implementation, routers check for the interface that is in the same VRF as the ingress interface.

It is usually recommended to ensure that the uRPF is configured in the correct mode. The strict mode should be configured for symmetric routing and the loose mode for asymmetric routing. In enterprise network operations, however, a combination of both modes is used [1], [39].
Cisco Express Forwarding (CEF) must be enabled before running uRPF, it can be enabled with the ip cef global configuration command. The uRPF implementation is an interface-based configuration [1]:

```
ip verify unicast source reachable-via {rx | any} [allow-default]
[list]
```

rx: Specifies the Strict mode.
any: Specifies the Loose mode.
allow-default: When the source IP address is reachable through the default path, it allows the packet to be routed.
list: Adds ACL for packets to be checked with uRPF.

14.3 Control Plane Policy

Control Plane Policing (CoPP) is a Cisco proprietary software security feature developed to control the unnecessary traffic rate handled by the network device's CPU [1]. Network traffic is classified into different classes based on their type, and then CoPP is applied as a QoS policy to limit the amount of traffic [40].

CoPP can be implemented in the following 4 steps [1]:

- **1th Step**: Creating ACLs for matching and identifying traffic.
- **2nd Step** (Defining a traffic class): Configuring Class Maps with ACLs created in the first step:

```
#Class-map {match-all|match-any|type}
 #match access-group {acl-num| name acl-name}
```

- **3rd Step** (Defining a Service policy): Configuring Policy Maps to police the traffic rate by allowing or dropping in three predefined actions conform, exceed, and violate:

```
#policy-map policy-name
 #class class-name
  #police Target-Bit-rate conform-action transmit|drop exceed-
action trasnmit|drop violate-action transmit|drop
```

Conform-action: Action when the rate is less than conform burst.
Exceed-action: Action, when the rate is within conforming, can conform + burst.
Violate-action: Action when the rate is greater than conform + exceed burst. Usually, set to be transmitted for critical traffic (IPsec, Management, Routing, etc.)

- **4th Step**: Applying the Policy Maps to the control plane:

```
#control-plane
 #Service-policy {input|output} policy-name
```

Verifying the Policy for CoPP:

```
show policy-map control-plane input
```

14.4 Trouble ticket #22.1.3.1 (Troubleshoot uRPF)

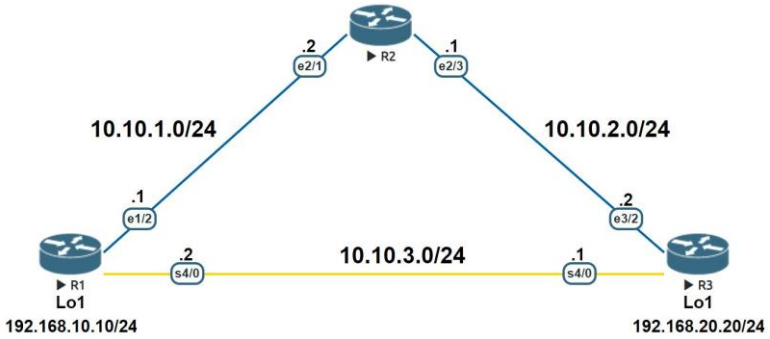


Figure 14. 1: Network Topology for the emulated lab "Troubleshoot uRPF"

In this section, trouble ticket #23.1.3.1 from “Troubleshoot labs” has been worked, on behalf of other trouble tickets.
All devices in Figure 14. 1 have an EIGRP adjacency with each other. It has been decided to implement uRPF on R1 to ensure network security against IP Spoofing attacks. However, after implementation, interface Lo1 on R3 lost its connection to resources in the 192.168.10.0/24 network.

```
R3#ping 192.168.10.10 source loopback 1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.10.10, timeout is 2 seconds:
Packet sent with a source address of 192.168.20.20
…..
Success rate is 0 percent (0/5)

R3#traceroute 192.168.10.10 source loopback 1
Type escape sequence to abort.
Tracing the route to 192.168.10.10
VRF info: (vrf in name/id, vrf out name/id)
  1  *  *  *
  2  *  *  *
  3  *  *  *
```

First of all, it needs to be verified that uRPF and CEF are enabled on the proper interface by using the show cef interface command for both interfaces e1/2 and s4/0:

```
R1#sh cef interface ethernet 1/2
Ethernet1/2 is up (if_number 9)
  …
  IP unicast RPF check is enabled
  …
  IP CEF switching enabled
  IP CEF switching turbo vector
```

The uRPF has been implemented in strict mode (rx) on interface ethernet1/2, so R1 only allows the packet which the source of the packet is reachable only via the ingress interface (Ethernet 1/2):

```
R1#show running-config | section Ethernet1/2
interface Ethernet1/2
 ip address 10.10.1.1 255.255.255.0
 ip verify unicast source reachable-via rx
```

According to the network topology in Figure 14. 1 R3 has another option to reach out to the resources of the 192.168.10.0/24 network. It can be implemented with a static route via serial4/0. The following show ip route command verifies that R3 routes via interface serial4/0 because the static route has a lower AD value than EIGRP. It means R1 receives the packet from R3 on interface s4/0 which is an unexpected interface for R1 and that is why R1 drops the packets:

```
R3#show ip route 192.168.10.10
Routing entry for 192.168.10.0/24
  Known via "static", distance 1, metric 0 (connected)
  Routing Descriptor Blocks:
  * directly connected, via Serial4/0
      Route metric is 0, traffic share count is 1
```

Static route on R3 must be deleted, then R3 will has only one option (EIGRP via e3/2) to reach out to resources on R1:

```
R3(config)#no ip route 192.168.10.0 255.255.255.0 Serial4/0


R3#show ip route
…
Gateway of last resort is not set
…
D     192.168.10.0/24 [90/435200] via 10.10.2.1, 00:00:08, Ethernet3/2

R3#traceroute 192.168.10.10 source loopback 1
Type escape sequence to abort.
Tracing the route to 192.168.10.10
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.2.1 0 msec 0 msec 0 msec
  2 10.10.1.1 1 msec *  3 msec
```

15. NETWORK MONITORING

Enterprise networks consist of hundreds of network devices and end systems such as printers, hosts, and servers, all of which may face problems with environmental conditions (fan alerts), capacity warning (CPU overutilization), infrastructure changes (link status). Network administrators can control the network over time with the information provided by network management protocols and tools. Network monitoring leads to active management and network optimization. This chapter covers the most common network management tools: **1)** SNMP, **2)** Syslog, **3)** IP SLA, and **4)** NetFlow.

The following two Troubleshooting labs are dedicated to the topics of this chapter:

- Lab **#23.1.3**_Troubleshoot SNMP and Logging Issue (It contains 2 trouble tickets 23.1.2.1-2).
- Lab **#23.1.4**_Troubleshoot IP SLA and NetFlow (It contains 3 trouble tickets 23.1.4.1-3).

All the laboratories are emulated and available in the EVE-NG server, devices have been configured with the initial configurations. Password on all devices is **cisco12345**, if a username is required, use the **admin**.

15.1 Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) is an application-layer protocol that was conceived in 1988 with RFC 1065. SNMP was developed for IP network management by gathering information about network device problems, end-system performance statistics, and network device configuration that is rarely used in configuration today [5].
SNMP has a Client-Server architecture, the server-side consists of an SNMP manager with Network Management Station (NMS) software running on the server, and the client-side includes an SNMP agent and Management Information Base (MIB) [1], [7].
The SNMP agent is a type of software on network devices/end systems that are monitored (CISCO IOS has the functionality of the SNMP agent on the devices, only need to be enabled with command lines). The MIB is a database for variables on the SNMP agent. The SNMP agent collects all information related to any changes, errors, and device performance then stores them in the MIB.
The SNMP manager uses UDP port 161 for sending a request to the SNMP agent, and the SNMP agent uses UDP port 162 to send any messages to the SNMP manager [1].
The following is a list of SNMP messages exchanged between the SNMP manager and agent [5], [7]:

- **SNMP Get**: This message is used by the NMS for polling the MIB to retrieve data. SNMP Get messages are sent automatically at preset intervals.

- **SNMP Set**: This message is used by the NMS to change the MIB and initiate an action such as rebooting the device.
- **SNMP Response**: This message is used by the SNMP agent in response to SNMP Get or SNMP Set messages.
- **SNMP Trap**: This type of message is an unsolicited message generated by the SNMP agent in response to any errors and thresholds. SNMP Trap messages are transmitted to the NMS.
- **SNMP Inform**: This type of message is the same as SNMP Trap with the difference in receiving an acknowledgment from the NMS. This type of message is supported only by SNMPv3.

There are 3 versions of this protocol: SNMPv1, SNMPv2c, and SNMPv3 [1]. The first version of SNMP is not widely used today, then SNMPv2 with extensive capabilities such as new SNMP Protocol Data Units (PDU), new MIB support, GetBulkRequest (used by NMS to retrieve large data blocks), and 64-bit variable counters vs 32-bit counters in SNMPv1 [1]. SNMPv1 and SNMPv2 both use the community string and access-list as the authentication method to ensure that the NMS access request to MIB is valid [1], [7]. There are two types of community strings: a) Read Only (RO), and b) Read-Write (RW) [5]. Compared to SNMP v1 and v2 versions, SNMPv3 provides more secure access to MIB by authenticating packets with usernames, MD5, and SHA algorithms, and improved the security level by encrypting packets with DES and AES algorithms over the networks [1], [5]. The following describes the steps for activating the SNMP agent on Cisco devices (SNMPv2) [1]:

- Create an ACL to identify authorized SNMP Management Stations.
- Define community strings that allow RO / RW access to MIB:

```
snmp-server community string ro | rw access-list-number-or-
name
```

- Configure SNMP agent with SNMP manager IP address where to send the SNMP traps:

```
snmp-server host host-id [traps | informs] [version {1| 2c | 3
[auth | noauth | priv]}] community-string [udp-port port-number]
[notification-type]
```

- Enabling the traps which will be sent by the SNMP agent:

```
snmp-server enable traps notification-types
```

15.2 Syslog

System Logging (SYSLOG) is a system logging protocol standardized in RFC-5424. Network devices use the Syslog protocol to send their event message (UDP #514) to a

server called Syslog Server, although these event messages are stored locally in the device memory [1].
Using SNMP, information can be retrieved from network devices, which is usually done sequentially and on schedule. In this case, the monitoring software sends a request to the device at short intervals and requests information such as disk space, free memory. Compared to SNMP, it is the device in Syslog that sends the event message without any request.
Each Syslog message can be up to 1024 bytes in size, and has the following information: Seq number, timestamp, event messages, and severity which are mandatory [1] [24], and optional information such as the host IP address, and diagnostic information, etc [1]. Each Syslog message consists of 3 parts: **1)** header (Syslog priority value + version), **2)** structured data, **3)** message. Syslog messages have specific priority/severity levels that can be modified.

The following Table 15. 1 is a list of the default priority level of each Syslog message [1]:

Level Keyword	**Level**	**Description**	**Syslog Definition**
Emergencies	0	System unstable	LOG_EMERG
Alerts	1	Immediate action needed	LOG_ALERT
Critical	2	Critical conditions	LOG_CRIT
Errors	3	Error conditions	LOG_ERR
Warnings	4	Warning conditions	LOG_WARNING
Notifications	5	Normal but significant conditions	LOG_NOTICE
Informational	6	Informational messages only	LOG_INFO
Debugging	7	Debugging messages	LOG_DEBUG

Table 15. 1: Syslog Message Severity Levels

By default, Syslog messages are stored internally in the console although this can be modified and stored in the system buffer (buffer size is 8192 bytes by default) by enabling it with logging buffer *buffer_size severity_level* global configuration command [1], [24]. The Syslog messages can also be sent to an external Syslog server by logging host *ip-address* global configuration command. By default, the Syslog server receives informational messages and lower but this can be modified by logging trap *severity_level* global configuration command [1].

15.3 NetFlow

CISCO IOS NetFlow is a flow-based network traffic collection feature that provides network statistics. NetFlow consists of two components, **a)** NetFlow Data Capture and **b)** NetFlow Data Export, capturing and exporting of traffic statistics are done with these two

components. The captured data statistics can be stored locally in the device memory or can be sent to the NetFlow collector as software installed on a remote server [1], [2]. The following are configuration instructions for NetFlow on a Cisco device [1]:

```
(config)#ip flow-export version version-number (Specify NetFlow version which should be matched with NetFlow collector)

(config)#ip flow-export destination ip-address port-number (Identify the IP address of the NetFlow collector with its dedicated port number)

(config)#interface interface-type interface-number
 (config-if)#ip flow ingress (Configure NetFlow to capture incoming traffic).
 (config-if)#ip flow ingress (configure NetFlow to capture outgoing traffic).
```

15.4 IP SLA

IP SLA is a built-in IOS tool that provides end-to-end monitoring for various aspects of the network. IP SLA can monitor traffic for jitter, latency, Packet loss, resource availability, connectivity, Website or Server response, and Voice quality [1], [2].
The following is the configuration of IP SLA operations **a) ICMP echo**_test the reachability and **b) HTTP GET**_monitoring the HTTP destinations [1]:

```
#ip sla operation-number (One number for each IP SLA probe separately)

 (ICMP echo)
 #icmp-echo {destination-ip-address | destination-hostname} [source-ip {ip-address | hostname} | source interface interface-name]
 #frequency seconds


 (HTTP GET)
 #http {get | raw} url [name-server ip-address] [version version-number] [source-ip {ip-address | hostname}] [source-port port-number] [cache {enable | disable}] [proxy proxy-url].

(Schedule and activate the IP SLA operations)
 #ip sla schedule operation-number [life {forever | seconds}] [start-time {[hh:mm:ss] [month day | day month] | pending | now | after hh:mm:ss}] [ageout seconds] [recurring]
```

When an IP SLA probe is implemented, it can be monitored using the CISCO-RTTMON-MIB file with SNMP, the traps are sent via Syslog to an SNMP server (NMS).

15.5 Troubleshooting

This section covers Cisco recommendations for troubleshooting network issues related to the network management tools. In addition, trouble ticket #23.1.3.1 from "Troubleshoot labs" has been worked at the end of this section, on behalf of other trouble tickets.

SNMP:

The following are Cisco recommendations for troubleshooting SNMP network issues [1]:

- The SNMP manager and agent must be able to ping each other.
- Ensure that there is no ACL to block port numbers 161 and 162.
- Ensure correct ACL configuration for authorized SNMP management stations.
- The Community strings should be matched between the NMS and agent.
- Incorrect configuration parameters for notifications such as trap activation, NMS IP address, and version specification.

SYSLOG:

The following are Cisco recommendations for troubleshooting network issues related to the Syslog [1]:

- The show logging command is used to verify the Syslog configuration and the Syslog messages stored in the system buffer.
- By default, the console and system buffer show debugging messages and lower.
- Ensure that no ACL denies UDP port 514.
- Old event messages are removed as the buffer size reaches the threshold level.
- The terminal monitor command must be enabled to view event messages when they reach the device via Telnet or SSH.
- It is recommended to use the NTP protocol to have an exact date and time.
- It is recommended to be enabled the timestamps on the Syslog message:

```
service timestamps [debug | log] [datetime | uptime])
```

NetFlow:

To prevent any NetFlow network problems, Cisco recommends that configure NetFlow parameters with a proper understanding of them. These parameters can be traffic direction, NetFlow enabled interface, Export source and destination, and the NetFlow version. The following are the most useful verification commands for NetFlow troubleshooting [1]:

```
#show ip flow interface (Verify the NetFlow enabled interfaces)
(Confirm NetFlow collector IP address and export statistics, including
errors):
#show ip flow export
#show ip cache flow (Verify the captured traffic flows)
```

IP SLA:

The IP SLA consists of an IP SLA source and IP SLA responder, the IP SLA source is a mandatory part of the IP SLA. Regarding the troubleshooting, CISCO recommends that IP addresses for both IP SLA source and responder must have reachability, it is necessary to select the correct port number and the correct IP SLA operation for the intended purpose of monitoring. The following are the most useful verification commands for IP SLA troubleshooting [1]:

```
#show ip sla to verify the scheduled IP SLA operation.
#show ip sla application to confirm which operations are supported and currently configured.
#show ip sla configuration to check the configured parameters for each IP SLA operation
#show ip sla responder to verify the operation of the IP SLA responder
```

15.5.1 Trouble ticket 23.1.3.1 (Troubleshoot SNMP and logging issue)

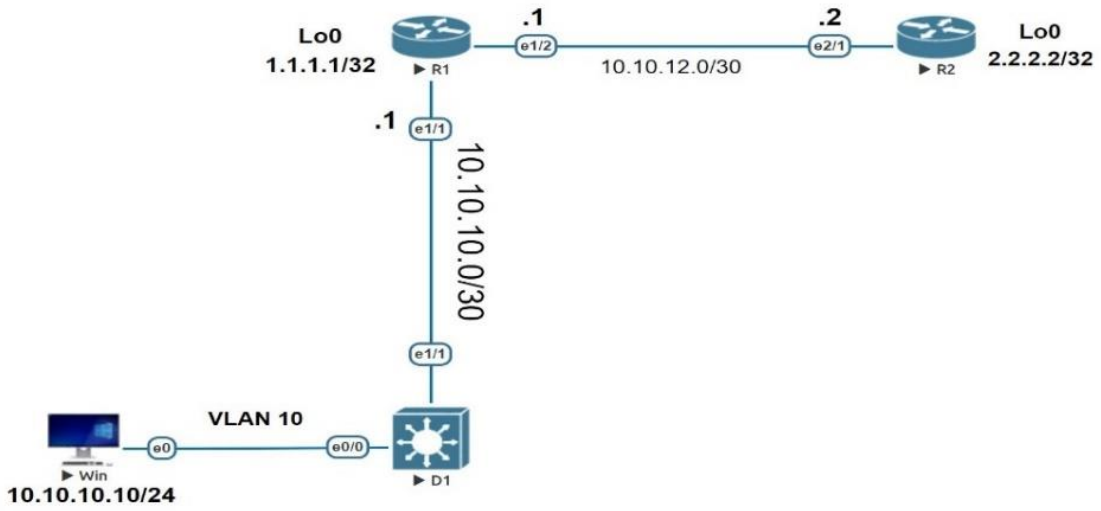


Figure 15. 1: Network topology for the emulated lab "Troubleshoot SNMP and logging issue"

All devices in this lab are configured with logging and SNMP features, KIWI Syslog server as the NMS is installed on PC (10.10.10.10) based on the provided settings in the Lab. In this lab, NMS should be received the traps both from R1 and D1 by enabling interface Lo0 but it only works for R1, see Figure 15. 2:

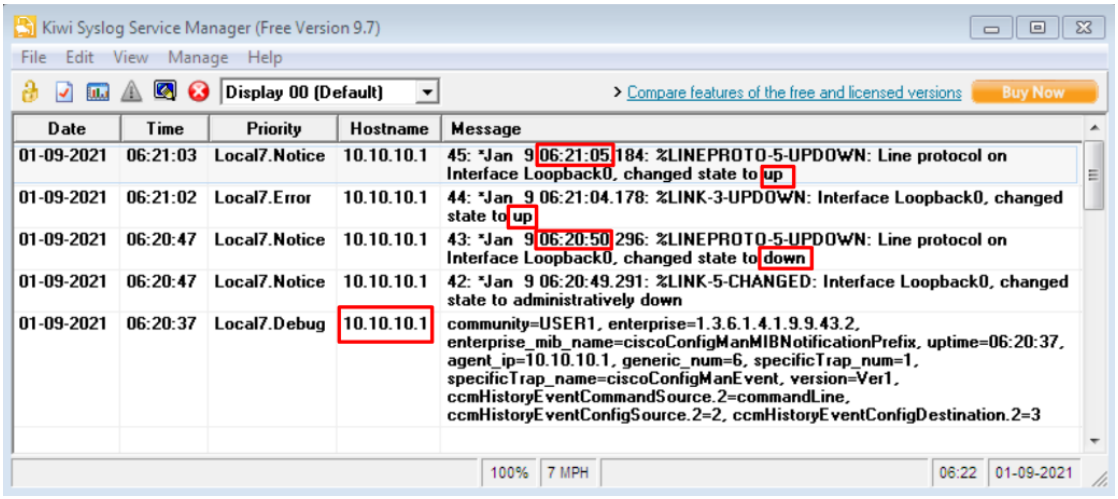


Figure 15. 2: Traps received only from R1 on NMS (KIWI Syslog server) installed in PC1.

According to Cisco recommendations, it may be due to incorrect ACL configuration, no IP connection, mismatched community string, or incorrect configuration parameters for notifications. We assume no IP connection issue and according to the following output there is no ACL is configured to block any related port or packet:

```
R1#sh ip access-lists
Standard IP access list PERMIT-ADMIN
    10 permit 10.10.10.0, wildcard bits 0.0.0.255
Extended IP access list ALLOW-TFTP
    10 permit udp 10.10.0.0 0.0.255.255 host 10.10.10.10 eq tftp
    20 permit udp 10.10.0.0 0.0.255.255 host 10.10.10.10 gt 1024
    30 permit udp 10.10.0.0 0.0.255.255 host 10.10.10.10 eq syslog
    40 permit udp 10.10.0.0 0.0.255.255 host 10.10.10.10 eq snmp
    50 permit udp 10.10.0.0 0.0.255.255 host 10.10.10.10 eq snmptrap
    60 permit tcp 10.10.0.0 0.0.255.255 eq 22 host 10.10.10.10
    70 permit icmp any any

D1#sh ip access-lists
Standard IP access list SNMP_ACL
    10 permit 10.10.10.10
```

It is time to check the configured SNMP parameters on the SNMP server (KIWI Syslog server on PC1) and SNMP client (D1). The following Figure 15. 3 is an example of the parameters configured on the SNMP server and we assume that the other parameters are also configured correctly:

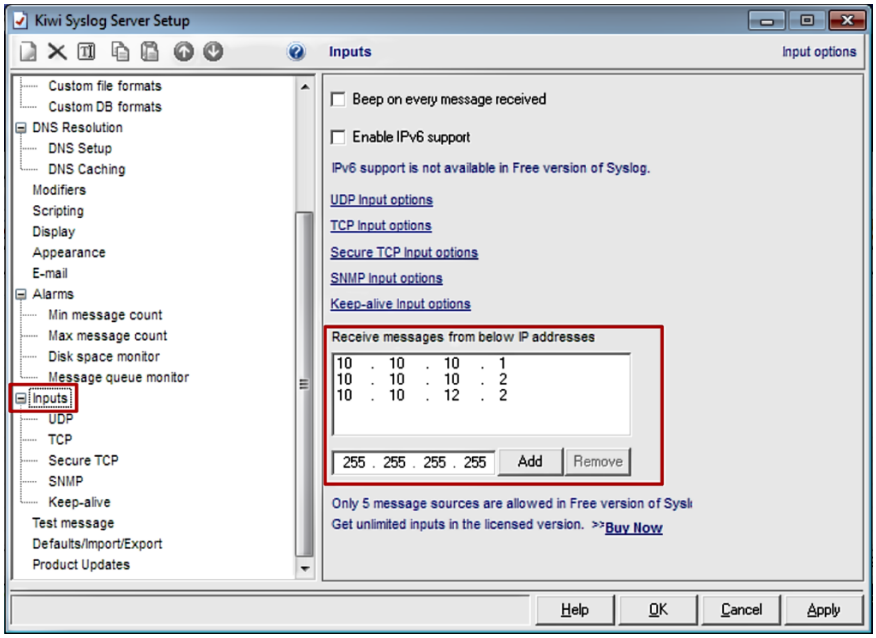


Figure 15. 3: Adding the SNMP client IP addresses in NMS.

The show snmp host command is used to verify the SNMP server (receiver) details for SNMP operations:

```
D1#show snmp host
Notification host: 10.10.10.11  udp-port: 162   type: trap
user: ciscolab  security model: v2c
```

According to the above output, the SNMP client (D1) has been configured with an incorrect host IP address and it should be configured as the following:

```
D1(config)#no snmp-server host 10.10.10.11 version 2c ciscolab
D1(config)#snmp-server host 10.10.10.10 version 2c ciscolab
```

Receiving the traps from D1 after identifying the correct authorized SNMP Management Stations:

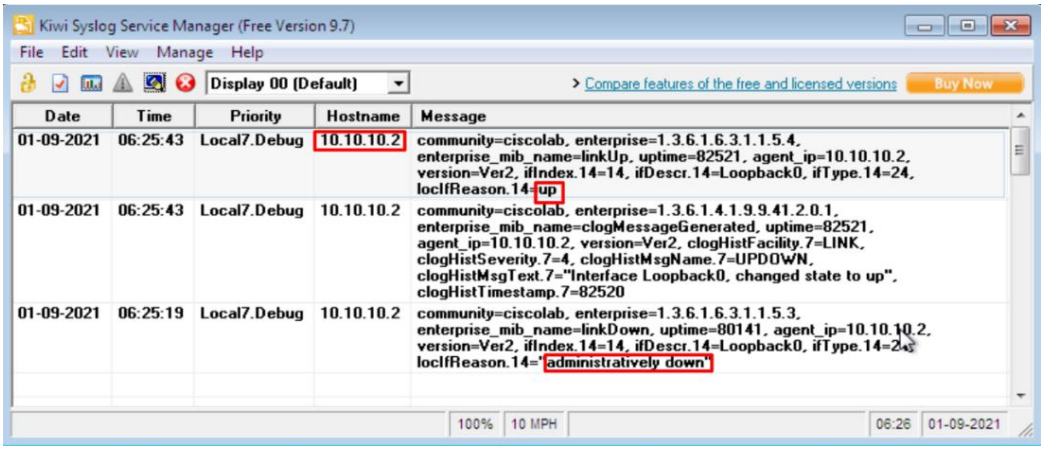


Figure 15. 4: Traps received from D1 in NMS after troubleshooting.

16. CONCLUSION

The aim of this thesis was primarily to get acquainted with the topics of laboratory tasks for the new CISCO certificate, CCNP Enterprise Advanced Routing and Services (ENARSI).

The following steps in the process are required, to compare possible simulation environments and then select the most suitable environment in which the laboratory tasks will be simulated. It is also necessary to create a topology for each laboratory task and a configuration file (.txt) for Troubleshooting tasks, then lastly upload the file to the appropriate device in the topology. The final step of the thesis was to design and implement a solution for remote access to prepared laboratories. All the set-out goals of this bachelor thesis were greatly achieved.

The whole part of the theory was studied and understood using different resources; an emulation environment was selected as the most suitable one and 40 laboratories were emulated inside it which consist of a topology of the network and an associated configuration file uploaded to each device. Finally, a remote access solution was designed and implemented for the prepared laboratories.

The first chapter highlighted that we are observing the emulation of the laboratory tasks and not the simulation. While these two words are used interchangeably, the emulation of laboratories was the task of the thesis since an emulator replicates every function of the system as much as possible until it gets an exact copy of that system which leads to more realistic results. Furthermore, two possible free emulation environments (GNS3 vs EVE-NG) have been analyzed according to the remote access solution. It is found that only two ways are possible to get access to prepared labs remotely in GNS3: a) Installing GNS3 on a remote server that uses the OpenVPN software b) Using GNS3 Web-UI with a separately designed VPN solution.

The EVE-NG environment has been selected as the most suitable environment, as GNS3 has a buggy and less user-friendly Web-UI, and has an unstable connection with the GNS3 installation on the remote server that uses the OpenVPN software.

Chapter 2 described how to set up an EVE-NG server with the appropriate emulators. Qemu was chosen for the operating system host but due to the lack of support for CLI, it has been decided to use the IOL for the Cisco L2/L3 devices.

The most exciting element of this thesis was the design and implementation of a remote access solution for laboratory tasks prepared at EVE-NG. It consists of solving the problem by implementing an L2TP/IPsec VPN solution between any L2TP/IPsec supported OS and Mikrotik router as an L2TP/IPsec server.

To better understand the topics of Chapters 4 to 15, I have referred to various resources and materials, reading and referencing, and internet resources.

The emulation of laboratories had no issues, all the labs were emulated and available on the EVE-NG server.

Regarding the remote access solution, the server is configured for two users' access with different credentials, see in Figure 3. 7, with two simultaneous users access, as soon as the second user accesses the server, the connection for the first user is lost. This is due to the lack of support in the EVE-NG Community version; however, this problem can be solved by using the EVE-NG Pro version.

17. LITERATURE

[1] LACOSTE, Raymond and Brad EDGEWORTH. CCNP enterprise advanced routing: ENARSI 300-410. Hoboken: Cisco Press, 2020. ISBN 978-1-58714-525-4.

[2] EDGEWORTH, Brad, Ramiro Garza RIOS, Jason GOOLEY and Dave HUCABY. CCNP and CCIE enterprise core: ENCOR 350-401. Hoboken: Cisco Press, 2020. ISBN 978-1-58714-523-0.

[3] JEŘÁBEK, Jan. Komunikační technologie. verze 2020. Brno: Vysoké učení technické v Brně, 2014. ISBN 978-80-214-4713-4. (CS)

[4] Emulation-vs-Simulation. Jadi.net [online]. Jadinet, 2016 [cit. 2021-2-9]. Dostupné z: https://jadi.net/2016/06/emulation-vs-simulation/

[5] TEARE, Diane. Implementing Cisco IP routing (ROUTE) foundation learning guide. Indianapolis, IN: Cisco Press, c [2015]. ISBN 978-1-58720-456-2.

[6] RFC 2328: OSPF Version 2 [online]. USA: RFC Editor, 1998 [cit. 2021-5-15]. Dostupné z: http://www.ietf.org/rfc/rfc2328.txt

[7] Connecting networks v6: companion guide. Indianapolis, IN: Cisco press, [2018]. ISBN 978-1-58713-432-6.

[8] Configure Single-Area OSPFv3: Similarities Between OSPFv2 and OSPFv3. CISCO NETWORKING ACADEMY (NUM) [online]. [cit. 2021-2-24]. Dostupné z: http://cisco.num.edu.mn/CCNA_R&S2/course/module8/8.3.1.2/8.3.1.2.html

[9] Extended ACL Configuration Commands Explained. ComputerNetworkingNotes [online]. 2018 [cit. 2021-3-27]. Dostupné z: https://www.computernetworkingnotes.com/ccna-study-guide/extended-acl-configuration-commands-explained.html

[10] Troubleshooting Methods for Cisco IP Networks. CiscoPress [online]. USA: Cisco Press, 2015 [cit. 2021-3-29]. Dostupné z: https://www.ciscopress.com/articles/article.asp?p=2273070&seqNum=2S

[11] OSPF Neighbor States. Cisco.com [online]. Cisco, 2014 [cit. 2021-4-12]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13685-13.html

[12] What Is Administrative Distance? Cisco.com [online]. Cisco, 2020 [cit. 2021-5-14]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/15986-admin-distance.html

[13] OSPF Inter-Area Routing. Cisco.com [online]. Cisco, 2005 [cit. 2021-5-17]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/47864-ospfdb5.html

[14] Initial Configurations for OSPF over a Point-to-Point Link. Cisco.com [online]. Cisco, 2007 [cit. 2021-5-12]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13687-15.html

[15] How Does OSPF Generate Default Routes? Cisco.com [online]. Cisco, 2005 [cit. 2021-4-19]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13692-21.html

[16] OSPF Design Guide. Cisco.com [online]. Cisco, 2005 [cit. 2021-4-22]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html#t6

[17] Sample Configuration for OSPFv3. Cisco.com [online]. Cisco, 2010 [cit. 2021-4-26]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/ip-version-6-ipv6/112100-ospfv3-config-guide.html

[18] IPv6 Routing: OSPFv3. Cisco.com [online]. Cisco, 2012 [cit. 2021-4-27]. Dostupné z: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/15-1sg/ip6-route-ospfv3.html

[19] Troubleshooting Duplicate Router IDs with OSPF. Cisco.com [online]. Cisco, 2012 [cit. 2021-3-22]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/23862-duplicate-router-id-ospf.html

[20] DOYLE, Jeff and Jennifer CARROLL. CCIE professional development routing TCP / IP. 2nd ed. Indianapolis: Cisco Press, 2006. ISBN 1-58705-202-4.

[21] Redistributing Routing Protocols. Cisco.com [online]. Cisco, 2012 [cit. 2021-3-23]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/8606-redist.html

[22] Understanding Redistribution of OSPF Routes into BGP. Cisco.com [online]. Cisco, 2012 [cit. 2021-3-24]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5242-bgp-ospf-redis.html

[23] VRF Lite. Networkdirection.net [online]. [cit. 2021-4-2]. Dostupné z: https://networkdirection.net/articles/routingandswitching/vrflite/

[24] System Message Logging. Cisco.com [online]. Cisco [cit. 2021-3-18]. Available from: https://www.cisco.com/c/en/us/td/docs/routers/access/wireless/software/guide/SysMsgLogging.html

[25] Ip prefix-list. Cisco.com [online]. Cisco [cit. 2021-4-27]. Dostupné z: https://www.cisco.com/c/m/en_us/techdoc/dc/reference/cli/nxos/commands/bgp/ip-prefix-list.html

[26] Using the Router as a TFTP Server. Oreilly.com [online]. oreilly [cit. 2021-4-13]. Dostupné z: https://www.oreilly.com/library/view/cisco-ios-cookbook/0596527225/ch01s14.html

[27] LAMMLE, Todd. CCNA Certification Study Guide, Volume 2: Exam 200-301 . USA: John Wiley, 2020. ISBN 978-1-119-65918-1.

[28] LAMMLE, Todd. CCNA Certification Study Guide, Volume 1: Exam 200-301 . USA: John Wiley, 2020. ISBN 978-1-119-65902-0.

[29] RFC 2460: Internet Protocol, Version 6 (IPv6) Specification [online]. USA: RFC Eitor, 1998 [cit. 2021-2-19]. Dostupné z: https://www.ietf.org/rfc/rfc2460.txt

[30] Configure EIGRP Named Mode. Cisco.com [online]. Cisco, 2017 [cit. 2021-3-13]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/200156-Configure-EIGRP-Named-Mode.html

[31] EIGRP Message Authentication Configuration Example. Cisco.com [online]. Cisco, 2007 [cit. 2021-3-16]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/82110-eigrp-authentication.html

[32] EIGRP IPv6 Configuration Example. Cisco.com [online]. Cisco, 2016 [cit. 2021-3-19]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/113267-eigrp-ipv6-00.html

[33] How Does Load Balancing Work? Cisco.com [online]. Cisco, 2015 [cit. 2021-3-24]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html

[34] How Does Unequal Cost Path Load Balancing (Variance) Work in IGRP and EIGRP? Cisco.com [online]. Cisco, 2009 [cit. 2021-3-24]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13677-19.html

[35] How Does the Passive Interface Feature Work in EIGRP? Cisco.com [online]. Cisco, 2021 [cit. 2021-3-18]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13675-16.html

[36] Introduction to EIGRP. Cisco.com [online]. Cisco, 2005 [cit. 2021-3-18]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13669-1.html#neighbor_table

[37] Cisco DMVPN: Choosing The Right Routing Protocol. Networkcomputing [online]. 2015 [cit. 2021-5-5]. Dostupné z: https://www.networkcomputing.com/networking/cisco-dmvpn-choosing-right-routing-protocol

[38] Dynamic Multipoint VPN Configuration Guide, Cisco IOS Release 15M&T. Cisco.com [online]. Cisco [cit. 2021-3-15]. Dostupné z: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_dmvpn/configuration/15-mt/sec-conn-dmvpn-15-mt-book/sec-conn-dmvpn-dmvpn.html

[39] Understanding Unicast Reverse Path Forwarding. Cisco.com [online]. Cisco [cit. 2021-5-1]. Dostupné z: https://tools.cisco.com/security/center/resources/unicast_reverse_path_forwarding

[40] CoPP?! What is that? CiscoZine.com [online]. CiscoZine, 2009 [cit. 2021-4-28]. Dostupné z: https://www.ciscozine.com/copp-what-is-that/

[41] GNS3: Documentation [online]. docs.gns3.com [cit. 2021-2-10]. Dostupné z: https://docs.gns3.com/docs/

[42] Emulator. Wikipedia.org [online]. Wikipedia [cit. 2021-2-10]. Dostupné z: https://en.wikipedia.org/wiki/Emulator

[43] Dynamips. Wikipedia.org [online]. Wikipedia [cit. 2021-2-10]. Dostupné z: https://en.wikipedia.org/wiki/Dynamips

[44] EVE-NG: Documentation [online]. eve-ng.net [cit. 2021-2-15]. Dostupné z: https://www.eve-ng.net/index.php/documentation/

[45] Main Page [online]. wiki.qemu.org [cit. 2021-2-11]. Dostupné z: https://wiki.qemu.org/Main_Page

[46] GNS3: Web-Ui Feedback [online]. gns3.com [cit. 2021-2-11]. Dostupné z: https://gns3.com/community/featured/gns3-web-ui-feedback

[47] EVE-NG: Features [online]. eve-ng.net [cit. 2021-2-15]. Dostupné z: https://www.eve-ng.net/index.php/features-compare/

[48] EVE-NG: community-cookbook [online]. eve-ng.net [cit. 2021-2-15]. Dostupné z: https://www.eve-ng.net/index.php/documentation/community-cookbook/

[49] Manual: Interface / L2TP. Mikrotik Documentation [online]. wiki.mikrotik.com [cit. 2021-4-20]. Available from: https://wiki.mikrotik.com/wiki/Manual:Interface/L2TP#Summary

[50] What can L2TP do for your network? NetworkWorld [online]. www.networkworld.com, 2013 [cit. 2021-4-20]. Dostupné z: https://www.networkworld.com/article/2163334/what-can-l2tp-do-for-your-network-.html

[51] DOYLE, Jeff. Routing TCP / IP, Volume II: CCIE Professional Development. Second Edition. Indianapolis: Cisco Press, 2017. ISBN 1-58705-202-4.

[52] Sample Configuration for iBGP and eBGP With or Without a Loopback Address. Cisco.com [online]. Cisco [cit. 2021-3-7]. Dostupné z: https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13751-23.html

[53] Configuring IKE Policies. Cisco.com [online]. Cisco [cit. 2021-3-19]. Dostupné z: https://www.cisco.com/assets/sol/sb/isa500_emulator/help/guide/ab1439828.html

[54] About IPSec VPN Negotiations. Watchguard.com [online]. watchguard [cit. 2021-3-18]. Dostupné z: https://www.watchguard.com/help/docs/help-center/en-US/Content/en-US/Fireware/mvpn/general/ipsec_vpn_negotiations_c.html#:~:text=To%20build%20the%20VPN%20tunnel,other%20device%20is%20the%20responder.

18. LIST OF SYMBOLS

AAA	Authentication, Authorization, and Accounting
ABR	Area Border Router
ACL	Access Control List
AD	Administrative Distance
AF	Address Family
AH	Authentication Header
ASBR	Autonomous System Boundary Router
ASN	Autonomous System Number
BDR	Backup Designated Router
BGP	Border Gateway Protocol
CEF	Cisco Express Forwarding
CoPP	Control Plane Policing
DHCP	Dynamic Host Configuration Protocol
DMVPN	Dynamic Multipoint Virtual Private Network
DR	Designated Router
DUAL	Diffusing Update Algorithm
EAP	Extensible Authentication Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
ESP	Encapsulation Security Payload
EVE-NG	Emulated Virtual Environment – Next Generation
FD	Feasible Distance
FSM	Finite-State Machine
GNS-3	Graphical Network Simulator-3
GRE	Generic Routing Encapsulation
IKE	Internet Key Exchange
IOL/U	IOS on Linux/Unix
IPsec	IP security
ISAKMP	Internet Security Association & Key Management Protocol
L2TP	Layer 2 Tunneling Protocol
LSA	Link-state Advertisement
LSDB	Link-State Database
MD5	Message-Digest 5
MIB	Management Information Base
MP-BGP	MultiProtocol BGP
MSS	Maximum Segment Size
MTU	Maximum Transmit Unit
NA	Neighbor Advertisement

NHC	Next Hop Client
NHRP	Next Hop Resolution Protocol
NHS	Next Hop Server
NLRI	Network Layer Reachability Information
NMS	Network Management Station
NS	Neighbor Solicitation
NSSA	Not-So-Stubby Area
OSPF	Open Shortest Path First
PA	Path Attribute
PBR	Policy-Base Routing
RADIUS	Remote Authentication Dial-In User Service
RD	Reported Distance
RS	Router Solicitation
SAFI	Subsequent Address Family Identifier
SLA	Service Level Agreement
SLAAC	Stateless Address Autoconfiguration
SNMP	Simple Network Management Protocol
SYSLOG	System Logging
TACACS+	Terminal Access Controller Access-Control System Plus
TFTP	Trivial File Transfer Protocol
uRPF	unicast Reverse Path Forwarding
VIRL	Virtual Internet Routing Lab
VLSM	Variable Length Subnet Mask
VPN	Virtual Private Network
VRF	Virtual routing and forwarding

19. LIST OF APPENDICES

Appendix A: Content of the CD

The CD contains one folder for each lab, for a total of **40 folders**. The Laboratories consist of two types of **Implementation** and **Troubleshooting**.

The implementation labs folder consists of:

- Configuration files (**.txt**) in the Config_Files folder,
- EVE-NG file (**.unl**) that is exported/imported in a compressed file (**.zip**),
- Experiment_File (**.docx**): Labs have been worked on to verify that all commands are executable on IOL/U images, which makes it possible to compare the emulated outputs with the original data.

The troubleshooting labs folder consists of:

- Dedicated folder for each trouble ticket, that consists of:
  - Configuration files (**.txt**),
  - EVE-NG file (**.unl**),
  - Experiment_File (**.docx**).